



SISTEM KENDALI NIRKABEL ROBOT BULUTANGKIS

BERBASIS MIKROKONTROLLER

PROYEK AKHIR

Diajukan Kepada Fakultas Teknik Universitas Negeri Yogyakarta

Untuk Memenuhi Sebagian Persyaratan

Guna Memperoleh Gelar Ahli Madya Teknik



Oleh:

Muhamad Shobirin

NIM. 13507134006

PROGRAM STUDI TEKNIK ELEKTRONIKA D3

FAKULTAS TEKNIK

UNIVERSITAS NEGERI YOGYAKARTA

2016

PERSETUJUAN

Proyek akhir yang berjudul “**Sistem Kendali Nirkabel Robot Bulutangkis Berbasis Mikrocontroller**” ini telah disetujui oleh pembimbing untuk diujikan.



Yogyakarta, 28 Juni 2016

Mengetahui,
Kaprodi Teknik Elektronika

Dr. Sri Waluyanti
NIP. 19581218 198603 2 001

Menyetujui,
Pembimbing Proyek Akhir


Nurkhamid, Ph. D.
NIP. 19680707 199702 1 001

PENGESAHAN

Proyek akhir yang berjudul “Sistem Kendali Nirkabel Robot Bulutangkis Berbasis Mikrokontroller” ini telah dipertahankan di depan Dewan Penguji pada tanggal 13 Juli 2016 dan dinyatakan LULUS.

DEWAN PENGUJI			
Nama	Jabatan	Tandatangan	Tanggal
Nurkhamid, Ph.D	Ketua Penguji		26/7/2016
Dr. Sri Waluyanti	Sekretaris Penguji		27/7/2016
Dr. Fatchul Arifin	Penguji		28/07-2016

Yogyakarta, 13 Juli 2016
Dekan Fakultas Teknik UNY


Dr. Widarto, M.Pd.
NIP. 19631230 198812 1 001

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam Proyek Akhir ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar ahli madya atau gelar lainnya di suatu perguruan tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis oleh orang lain, kecuali secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 28 Juni 2016
Yang menyatakan,



Muhamad Shobirin
NIM. 13507134006

MOTTO

*“ bersabar dan tetaplah bersabar menjalani kehidupan, yakinlah keberuntungan
dan keberhasilan senantiasa mengiringi ”*

(my self)

“Innallaha Ma’ash Shobirin”

(Al-Quran Surah Al-Baqarah ayat 153)

“Segala sesuatu bergantung pada niatnya ”

(Hadist Bukhari-Muslim)

*“Sesungguhnya Allah Tidak Akan Mengubah Nasib Suatu Kaum Kecuali Kaum
Itu Sendiri Yang Mengubah Apa-Apa Yang Pada Diri Mereka”*

(Al-Quran Surah Ar Ra’d ayat 11)

PERSEMBAHAN

Dengan penuh rasa syukur kepada Allah SWT, karya ini saya persembahkan kepada:

1. Kedua orang tua saya yang telah memberikan dorongan dan kasih sayangnya serta selalu mendoakan sepanjang waktu untuk keberhasilan saya.
2. Untuk keluarga besar yang selalu memberi dukungannya.
3. Kawan-kawan seperjuangan Tim Robot Universitas Negeri Yogyakarta yang telah memberikan pelajaran dan pengalaman yang sangat berharga serta kekeluargaan yang terjalin.
4. Untuk rekan-rekan jurusan Pendidikan Teknik Elektronika angkatan 2013 yang senantiasa berbagi ilmu.
5. Perpustakaan Universitas Negeri Yogyakarta untuk kemajuan ilmu pengetahuan dan teknologi.

Terima kasih atas segala doa, bimbingan, kasih sayang, dan pengorbanan yang tidak pernah berhenti.

ABSTRAK

Sistem Kendali Nirkabel Robot Bulutangkis Berbasis Mikrokontroller

Muhamad Shobirin

NIM. 13507134006

Era globalisasi menumbuhkan persaingan pasar bebas di bidang teknologi semakin meningkat. Robotika adalah salah satu teknologi yang sangat dibutuhkan. Negara Indonesia dibawah naungan KEMENRISTEK DIKTI menyelenggarakan Kontes Robot ABU Indonesia dengan tema Robot bulutangkis. Pembuatan proyek akhir yang berjudul Sistem Kendali Nirkabel Robot Bulutangkis Berbasis Mikrokontroller ini dibuat dari beberapa blok rangkaian yang fungsional dengan tujuan supaya mudah dalam perakitan dan pengoperasian sesuai dengan peraturan Kontes Robot ABU Indonesia.

Pembuatan Proyek Akhir ini terdiri dari beberapa tahapan yaitu identifikasi kebutuhan, analisis kebutuhan, perancangan rangkaian, langkah pembuatan alat, diagram alir program, perancangan program, pengujian alat dan pengambilan data. Pembuatan mekanik robot menggunakan bahan dasar *aluminium profile* yang dirangkai sedemikian rupa sesuai desain agar robot dapat melaksanakan tugasnya dengan baik. Titik berat robot dirancang serendah mungkin agar robot mampu bermanuver dengan lincah. Robot menggunakan tiga buah mikrokontroler AVR yang terdiri dari satu buah mikrokontroler Master, satu buah mikrokontroler slave pemroses data serial bluetooth dan satu buah mikrokontroler pemroses aktuator tambahan. Ketiga mikrokontroler dihubungkan dengan protokol komunikasi serial UART. Robot bulutangkis dirancang mampu bermanuver, menservis dan membalikkan kok. Servis kok, pengembalian servis dan manuver oleh robot bulutangkis dapat diatur secara *real time* menggunakan komunikasi nirkabel berupa bluetooth hc-05.

Hasil pengujian menunjukkan bahwa tahapan pembuatan robot bulutangkis dapat dirancang dan diimplementasikan dengan baik. Unjuk kerja sistem kendali nirkabel robot bulutangkis berbasis mikrokontroler menunjukkan bahwa robot mampu menjepit dan melepas kok, menservis, membalikkan servis, dan bermanuver dengan tingkat keberhasilan 100%.

Kata Kunci: *robot bulutangkis, mikrokontroler, bluetooth*

KATA PENGANTAR

Alhamdulillah rabbil'aalamiin, dengan penuh keikhlasan dan kerendahan hati, penulis menghaturkan syukur Alhamdulillah yang sebesar-besarnya kepada ALLAH SWT Yang Maha Pengasih dan Maha Penyayang yang senantiasa memberikan kekuatan, bimbingan, ampunan yang seluas- luasnya dan pertolongan dengan seagung-agungnya pertolongan, serta Sholawat dan Salam sepenuhnya kami haturkan kepada junjungan kami Baginda Nabi Muhammad SAW sebagai panutan dan penghulu kami sehingga kami dapat menyelesaikan proyek akhir kami yang berjudul "Sistem Kendali Nirkabel Robot Bulutangkis Berbasis Mikrokontroller" ini. Proyek akhir ini disusun guna melengkapi salah satu persyaratan untuk memperoleh gelar Ahli madya / diploma (D-3) Teknik Elektronika, Fakultas Teknik, Universitas Negeri Yogyakarta.

Tim Maestro_Evo terdiri dari Hermawan R.W. sebagai ketua tim, Khoirul P.R. sebagai driver robot 1 dan programmer 2, Haikal F.Z. sebagai driver robot 2 dan mekanik, Wendra B.S. sebagai programmer 1 dan mekanik, Cahya A.K. sebagai elektronik 1 dan mekanik, penulis sebagai elektronik 2 dan mekanik dan Intan R.D. selaku administrasi. Robot bulutangkis yang dibuat telah digunakan sebagai tugas akhir mata kuliah perancangan sistem otomasi industri (PSOI) oleh Wendra B.S. dengan judul rancang bangun robot forklif menggunakan empat *omni directional wheels* dan Khoirul P.R. dengan judul rancang bangun robot badminton menggunakan joystick. Penulis dengan konsentrasi elektronik membuat beberapa macam rangkaian elektronik yakni sistem minimum ATmega16, sistem minimum ATmega128, dan regulator 7805.

Terwujudnya Proyek Akhir ini tidak lepas dari bimbingan, saran, dan bantuan baik moril dan materiil, dorongan serta kritik dari beberapa pihak. Dengan hati yang tulus penulis sampaikan rasa terima kasih kepada:

1. Kedua orang tua dan keluarga yang selalu memberikan dorongan, semangat dan kasih sayangnya sehingga proyek akhir ini dapat diselesaikan.
2. Nurkhamid Ph.D selaku dosen pembimbing yang selalu memberikan masukan dan bimbingannya dalam mengerjakan proyek akhir.
3. Dr. Widarto selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta.
4. Dr. Fatchul Arifin selaku Ketua Jurusan Pendidikan Teknik Elektronika.
5. Dr. Sri Waluyanti selaku Ketua Program Studi Pendidikan Teknik Elektronika.
6. Teman seperjuangan Tim Robot UNY dan Segenap rekan tim MAESTRO_EVO
7. Untuk rekan-rekan program studi Teknik Elektronika kelas B jurusan Pendidikan Teknik Elektronika angkatan 2013 yang selalu bersama-sama dalam menjalani masa muda di kampus dengan semua canda dan tawa.
8. Semua Pihak yang telah membantu dan memberikan dorongannya sehingga proyek akhir ini dapat terselesaikan, semoga kebbaikannya menjadi amal ibadah.

Penulis menyadari bahwa dalam penyusunan laporan proyek akhir ini tidak lepas dari kesalahan dan kekurangan, maka kritik dan saran dari semua pihak, akan penulis terima demi kesempurnaan laporan ini. Penulis berharap semoga laporan ini bermanfaat bagi semua pihak yang akan membutuhkan.

Yogyakarta, 28 Juni 2016

Penulis

DAFTAR ISI

Halaman

PROYEK AKHIR	i
PERSETUJUAN	ii
PENGESAHAN	iii
PERNYATAAN.....	iv
MOTTO	v
PERSEMBAHAN	vi
ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN.....	xiv
BAB I PENDAHULUAN.....	1
A. Latar belakang.....	1
B. Identifikasi Masalah.....	3
C. Batasan Masalah	3
D. Rumusan Masalah.....	3
E. Tujuan Proyek Akhir.....	4
F. Manfaat Proyek Akhir.....	4
G. Keaslian Gagasan.....	5
BAB II KAJIAN TEORI.....	7
A. Sistem Kendali	7
B. Robot.....	16
C. Bulutangkis	17
D. <i>Wireless</i>	18
E. Mikrokontroler AVR (<i>Alf and Vegaard's Risc Processor</i>)	24
F. Mikrokontroler ATmega16.....	26
G. Fitur Mikrokontroler ATmega16	28
H. Blok Diagram Mikrokontroler ATmega16	32
I. Konfigurasi Pin AVR ATmega16.....	33
J. Status Register	34
K. Mikrokontroler ATmega128	36
L. Bahasa C	46
M. Perangkat Lunak CodeVision AVR (CVAVR)	48
N. UBEC	51
O. LCD <i>Display</i> 16x2 M1632.....	52
P. Motor Planetary Gear (PG) 45	53
Q. Pneumatik	53
BAB III PERANCANGAN DAN PEMBUATAN ALAT	57
A. Identifikasi Kebutuhan.....	57
B. Analisis Kebutuhan.....	58

C. Perancangan Rangkaian	62
D. Langkah Pembuatan Alat	65
E. Diagram Alir/ <i>Flowchart</i> Program.....	69
F. Perancangan Program	72
G. Perencanaan Pengujian dan Pengambilan Data	72
1. Pengujian catu daya.....	72
2. Pengujian pneumatik.....	73
3. Pengujian motor	74
4. Pengujian komunikasi <i>wireless</i> robot.....	74
BAB IV PENGUJIAN DAN PEMBAHASAN	76
A. Pengujian.....	76
1. Tujuan Pengujian	76
2. Tempat Pengujian.....	76
3. Hasil Pengujian	76
B. Pembahasan.....	79
A. Analisis pengujian catu daya.....	80
B. Analisis pengujian pneumatik	81
C. Analisis pengujian motor	81
D. Analisis pengujian komunikasi <i>wireless</i> robot	82
BAB V KESIMPULAN DAN SARAN.....	84
A. Kesimpulan	84
B. Keterbatasan Alat	85
C. Saran.....	85
DAFTAR PUSTAKA	86
LAMPIRAN.....	88

DAFTAR TABEL

	Halaman
Tabel 1. Fungsi Khusus Port A ATmega128	40
Tabel 2. Fungsi Khusus Port B ATmega128	40
Tabel 3. Fungsi Khusus Port C ATmega128	41
Tabel 4. Fungsi Khusus Port D ATmega128	41
Tabel 5. Fungsi Khusus Port E ATmega128.....	41
Tabel 6. Fungsi Khusus Port F ATmega128.....	42
Tabel 7. Fungsi Khusus Port G ATmega128	42
Tabel 8. Beberapa <i>compiler</i> C untuk mikrokontroler AVR.....	48
Tabel 9. Konfigurasi tombol pada joystick <i>wireless</i>	60
Tabel 10. Pengukuran tegangan regulator 7805 pada modul mikrokontroler ATmega16	73
Tabel 11. Pengukuran tegangan regulator 7805 pada modul mikrokontroler ATmega128	73
Tabel 12. Pengujian Pneumatik	73
Tabel 13. Pengujian Motor.....	74
Tabel 14. Pengujian Komunikasi <i>Wireless</i> Robot	75
Tabel 15. Hasil pengujian Pengukuran tegangan regulator 7805 pada modul mikrokontroler ATmega16.....	76
Tabel 16. Hasil pengujian Pengukuran tegangan regulator 7805 pada modul mikrokontroler ATmega128	77
Tabel 17. Hasil pengujian Pneumatik	77
Tabel 18. Hasil pengujian Motor	78
Tabel 19. Hasil pengujian komunikasi <i>wireless</i> robot.....	79

DAFTAR GAMBAR

	Halaman
Gambar 1. Deskripsi sederhana sistem kendali.....	7
Gambar 2. <i>Elevator response</i>	8
Gambar 3. Sistem Pengendalian lup tertutup	10
Gambar 4. Sistem pengendalian level cairan secara manual	11
Gambar 5. Sistem pengendalian level cairan secara otomatis	12
Gambar 6. Sistem pengendalian digital	13
Gambar 7. Sistem pengendalian kontinyu	16
Gambar 8. Blok diagram fungsional ATmega16	32
Gambar 9. Susunan Pin ATmega16	33
Gambar 10. Status register ATmega16	34
Gambar 11. Arsitektur Mikrokontroler ATmega128.....	37
Gambar 12. Blok Diagram Fungsional ATmega128	38
Gambar 13. Konfigurasi Pin ATmega128	39
Gambar 14. Memori Program AVR ATmega128.....	43
Gambar 15. Konfigurasi memori data AVR ATmega128	44
Gambar 16. Status Register AVR ATmega128	45
Gambar 17. IDE perangkat lunak CodeVisionAVR	50
Gambar 18. Kode generator yang dapat digunakan untuk menginisialisasi register-register pada mikrokontroler AVR	51
Gambar 19. Bentuk fisik UBEC	52
Gambar 20. LCD M1632	52
Gambar 21. Motor <i>planetary gear</i>	53
Gambar 22. Pneumatik.....	54
Gambar 23. Blok diagram Utama rangkaian.....	59
Gambar 24. Spesifikasi Blok diagram Utama rangkaian	59
Gambar 25. Rangkaian Sistem minimum ATmega128	62
Gambar 26. Rangkaian sistem minimum ATmega16.....	64
Gambar 27. Rancangan peletakan roda pada base robot.....	66
Gambar 28. Rancangan mekanik Robot.....	67
Gambar 29. Flowchart pengiriman data.....	70
Gambar 30. Flowchart penerima data	71

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Layout PCB Robot Bulutangkis	89
Lampiran 2. Program Sismin ATmega128	90
Lampiran 3. Program Sismin ATmega16	110
Lampiran 4. Gambar Desain Mekanik	117
Lampiran 5. Daftar komponen	118
Lampiran 6. Rangkaian elektronik penggerak motor <i>H-bridge</i>	119
Lampiran 7. Rangkaian elektronik penggerak pneumatik (driver pneumatik) ...	120

BAB I

PENDAHULUAN

A. Latar belakang

Inovasi di bidang teknologi instrumentasi dan kendali berkembang dengan cepat. Hal ini selaras dengan perkembangan karakteristik masyarakat yang memiliki mobilitas tinggi. Masyarakat kini menginginkan layanan yang fleksibel, mudah, memuaskan, efisien, dan aman tak terkecuali di bidang industri.

Pada era globalisasi seperti saat ini, persaingan pasar sudah sangat terbuka, hanyalah yang mampu bersaing yang dapat bertahan. Faktor kualitas menjadi jawaban. Industri yang lebih berkualitas dialah yang akan memenangkan persaingan. Sehingga suatu industri harus mampu menekan biaya produksi dan mengoptimalkan produksi.

Dalam industri yang berskala besar kegiatan produksinya sudah dilakukan secara otomatis yaitu dengan menggunakan robot yang bergerak otonom sebagai salah satu elemen sebagai proses manufaktur. Kehadiran robot-robot di dunia industri memberikan efek positif terhadap kemajuan industri tersebut. Dengan adanya proses otomatisasi industri, kinerja karyawan dapat berjalan secara efektif begitu juga dengan tingkat produksi suatu perusahaan.

Teknologi robot telah dikembangkan di berbagai negara di planet bumi ini, tidak terkecuali Indonesia. Sejak tahun 80-an, kebijakan nasional dalam pengembangan riset teknologi telah memberikan dukungan pada litbang permesinan otomatis dalam rangka menunjang Sumber Daya Manusia

Indonesia yang memiliki minat dan kemampuan untuk menguasai teknologi robot.

Wujud nyata dari aktivitas penelitian di bidang robotika di Indonesia adalah seperti yang dilakukan oleh para mahasiswa Perguruan Tinggi negeri ataupun swasta dibawah naungan KEMENRISTEK DIKTI yang berlomba-lomba membuat suatu robot yang handal untuk mengikuti perhelatan nasional tahunan. Setiap tahun di Indonesia diadakan perhelatan nasional Kontes Robot Indonesia (KRI) yang diikuti oleh mahasiswa Perguruan Tinggi di seluruh Indonesia. Kontes Robot Indonesia terbagi menjadi 4 kategori yaitu Kontes Robot Abu Indonesia (KRAI), Kontes Robot Sepak Bola Indonesia (KRSBI), Kontes Robot Pemadam Api Indonesia Beroda dan berkaki (KRPAI) dan kontes robot Seni Indonesia (KRSI). Pembuatan robot bulutangkis beracuan pada peraturan KRAI 2015 yakni robot dikendalikan dengan joystick *wireless* untuk menservis shuttlecock dan membalikkan servis lawan. Pembuatan robot disesuaikan dengan peraturan kontes. Keterbatasan dimensi robot biasanya menimbulkan kesulitan dalam perancangan sistem manipulasi robot. Robot harus berpindah tempat maka robot dirancang dengan fungsi mobile.

Banyak tantangan yang dihadapi dalam membuat robot bulutangkis, diantaranya adalah sejauh mana robot bisa dikendalikan dan dapat mengatasi gangguan koneksi data serta robot dapat menservis shuttlecock dan membalikkan servis lawan.

B. Identifikasi Masalah

Dari latar belakang masalah tersebut, maka dapat diidentifikasi beberapa masalah yang muncul sebagai berikut.

1. Belum adanya sistem manipulasi robot yang efektif diterapkan pada robot bulutangkis.
2. Belum adanya sistem kendali nirkabel robot bulutangkis yang efektif dan efisien untuk bergerak.
3. Belum adanya rancangan mekanik robot bulutangkis dengan dua raket.

C. Batasan Masalah

Berdasarkan identifikasi masalah yang muncul, perlu adanya pembatasan masalah sehingga ruang lingkup permasalahannya terfokus. Dalam proyek akhir ini akan membahas tentang sistem minimum ATmega16 sebagai pengirim data ke sistem utama dengan komunikasi nirkabel, sistem minimum ATmega128 sebagai penerima data dan pengeksekusi perintah ke aktuator, catu daya sebagai supply untuk sistem minimum ATmega16 dan sistem minimum ATmega128 yang digunakan dalam pembuatan sistem kendali nirkabel robot bulutangkis berbasis mikrokontroler via bluetooth.

D. Rumusan Masalah

Berdasarkan latar belakang masalah, identifikasi masalah serta batasan masalah, rumusan masalah yang akan dibahas adalah sebagai berikut.

1. Bagaimana pembuatan sistem kendali nirkabel robot bulutangkis berbasis mikrokontroler?
2. Bagaimana unjuk kerja Sistem Kendali Nirkabel Robot Bulutangkis?

E. Tujuan Proyek Akhir

Berdasarkan rumusan permasalahan, diharapkan akan dicapai tujuan sebagai berikut.

1. Membuat sistem kendali nirkabel robot bulutangkis berbasis mikrokontroller.
2. Mengetahui unjuk kerja robot bulutangkis dengan sistem kendali nirkabel.

F. Manfaat Proyek Akhir

Dari pembuatan sistem kendali nirkabel robot bulutangkis berbasis mikrokontroller ini dapat diperoleh beberapa manfaat bagi pihak-pihak terkait, manfaat proyek akhir sebagai berikut.

1. Bagi Institusi/Perguruan Tinggi
 - a. Sebagai referensi tambahan untuk mengembangkan teknologi robotika.
 - b. Sebagai media belajar disiplin ilmu robotika.
2. Bagi Mahasiswa
 - a. Memperoleh pemahaman dan pengetahuan dalam pengembangan fungsi robot di dunia industri.
 - b. Sebagai sarana untuk membantu mahasiswa dalam mengaplikasikan ilmu-ilmu yang telah didapat di bangku kuliah terutama di bidang otomasi dan robotika.
 - c. Memberikan motivasi kepada mahasiswa untuk tetap berkarya dan menjadi salah satu pelaku dalam kemajuan teknologi di jaman yang serba modern ini.

3. Bagi Industri

- a. Dapat dikembangkan sebagai robot pemindah barang dari suatu tempat ke tempat lain dengan komunikasi *wireless*.
- b. Akan terciptanya mesin-mesin semi otomatis dan otomatis yang dapat meringankan tugas manusia.

G. Keaslian Gagasan

Pembuatan Proyek akhir ini adalah sebagai riset yang digunakan pada Kontes Robot Abu Indonesia (KRAI) tahun 2015 dalam serangkaian Kontes Robot Indonesia (KRI) tahun 2015 mewakili almamater Universitas Negeri Yogyakarta. Dalam kontes yang digelar KEMENRISTEK DIKTI tersebut terdapat puluhan jenis robot bulutangkis yang dibuat oleh peserta. Robot-robot tersebut tentunya memiliki kekurangan dan kelebihan, diantaranya sebagai berikut.

1. Memiliki jumlah raket lebih dari 2, memudahkan untuk melakukan pengembalian servis. Namun berat robot akan bertambah dan gerakan robot melambat.
2. Menggunakan komunikasi nirkabel bluetooth yang tidak perlu diprogram sehingga ketika start langsung bisa terhubung. Namun saat pertandingan sering terjadi error.
3. Menggunakan robot otomatis dengan kamera sebagai penginderaan mata robot dengan aktuator penggerak raket berupa motor DC. Namun pencahayaan sangat berpengaruh terhadap gambar yang ditangkap kamera, ketika tidak sesuai maka robot akan error.

4. Kontroller yang digunakan arduino, mini pc, dan arm memiliki nilai clock tinggi serta bahasa pemrograman C++ dan C#. Namun ketika program error perlu waktu cukup lama untuk pembenahannya.

Adapun perbedaan kelebihan dan kekurangan yang terdapat dalam pembuatan proyek akhir ini adalah sebagai berikut.

1. Robot Bulutangkis memiliki dua buah raket. Raket pertama untuk servis dan membalikkan servis lawan, kemudian raket kedua untuk membalikkan servis lawan. Berat robot menjadi tidak terlalu berat dan robot dapat bergerak lebih cepat. Namun ketika ada gangguan udara dari AC ruangan robot kesulitan untuk mengembalikan servis lawan dikarenakan jumlah raket hanya 2.
2. Menggunakan komunikasi nirkabel bluetooth terprogram sehingga ketika start langsung bisa terhubung dan tidak error saat pertandingan.
3. Menggunakan robot semi otomatis saat servis dengan aktuator penggerak raket berupa motor DC dan aktuator pneumatik untuk menjepit kok. Namun ketika pertandingan servis robot agak melemah karena terkena AC ruangan dan menyebabkan lawan dengan mudah mengembalikan servis.
4. Kontroller yang digunakan ATmega16 dan ATmega memiliki nilai clock cukup tinggi yakni berkisar 12-16 MHz serta bahasa pemrograman adalah bahasa C dengan software CodeVision AVR. Ketika program error akan ada peringatan error secara otomatis sehingga mempercepat saat pembenahan program.

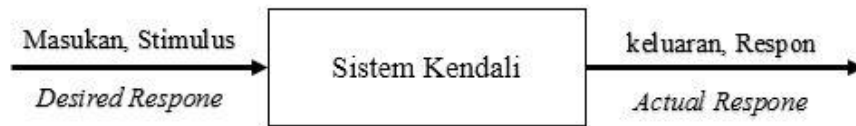
BAB II

KAJIAN TEORI

A. Sistem Kendali

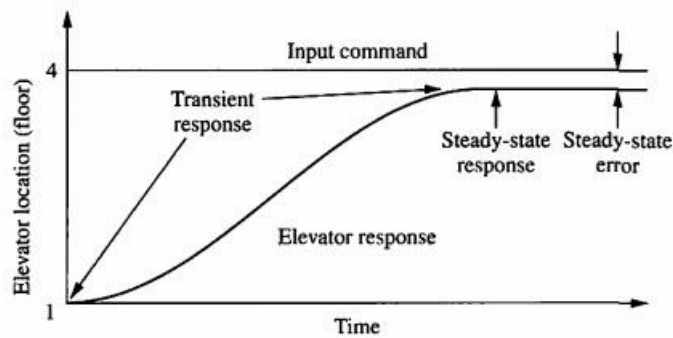
Sistem kendali adalah suatu susunan komponen fisik yang terhubung atau terkait sedemikian rupa sehingga dapat memerintah, mengarahkan, atau mengatur diri sendiri atau sistem lain. Di dalam dunia *engineering* dan *science* sistem kendali cenderung dimaksudkan untuk sistem kendali dinamis.

Sistem kendali terdiri dari sub-sistem dan proses yang disusun untuk mendapatkan keluaran dan kinerja yang diinginkan dari input yang diberikan. Gambar 1 menunjukkan blok diagram untuk sistem kendali paling sederhana, sistem kendali membuat sistem dengan input yang diberikan menghasilkan output yang diharapkan.



Gambar 1. Deskripsi sederhana sistem kendali
(Sumber : <http://www.eviandriani.com/2010/05/pengertian-sistem-kendali.html>)

Sebagai contoh, misalnya penggunaan *elevator* (lift), pada saat tombol yang menunjukkan nomor lantai tujuan ditekan, maka *elevator* akan bergerak naik/turun menuju lantai tujuan tersebut. Tombol bernomor lantai tujuan yang ditekan tersebut merupakan input yang menunjukkan output yang kita inginkan. Sistem ini merupakan fungsi step yang ditunjukkan pada Gambar 2, kinerja *elevator* dapat dilihat dari kurva *elevator response*.



Gambar 2. *Elevator response*
(Sumber : <http://www.eviandriani.com/2010/05/pengertian-sistem-kendali.html>)

Dua kinerja utama terukur yang dapat dilihat adalah, pertama, respons transient, kedua, *steady-state error*. Pada contoh *elevator* ini, kenyamanan dan waktu yang dibutuhkan untuk sampai pada tujuan pengguna bergantung pada respons transient. Jika respon ini terlalu cepat, kenyamanan penumpang yang dikorbankan, jika terlalu lambat, waktu yang diperlukan juga semakin besar. *Steady-state error* juga merupakan indikator kinerja yang sangat penting karena keselamatan penumpang dan kenyamanan akan dikorbankan jika output tidak sesuai yang diinginkan.

Istilah-istilah dalam sistem pengendalian sebagai berikut.

1. Masukan

Masukan atau input adalah rangsangan dari luar yang diterapkan ke sebuah sistem kendali untuk memperoleh tanggapan tertentu dari sistem pengaturan. Masukan juga sering disebut respon keluaran yang diharapkan.

2. Keluaran

Keluaran atau output adalah tanggapan sebenarnya yang didapatkan dari suatu sistem kendali.

3. *Plant*

Seperangkat peralatan atau objek fisik dimana variabel prosesnya akan dikendalikan, misalnya pabrik, reaktor nuklir, mobil, sepeda motor, pesawat terbang, pesawat tempur, kapal laut, kapal selam, mesin cuci, mesin pendingin (sistem AC, kulkas, *freezer*), penukar kalor (*heat exchanger*), bejana tekan (*pressure vessel*), robot dan sebagainya.

4. Proses

Berlangsungnya operasi pengendalian suatu variabel proses, misalnya proses kimiawi, fisika, biologi, ekonomi, dan sebagainya.

5. Sistem

Kombinasi atau kumpulan dari berbagai komponen yang bekerja secara bersama-sama untuk mencapai tujuan tertentu.

6. Diagram blok

Bentuk kotak persegi panjang yang digunakan untuk mempresentasikan model matematika dari sistem fisik. Contohnya adalah kotak pada Gambar 1 atau 2.

7. Fungsi Alih (*Transfer Function*)

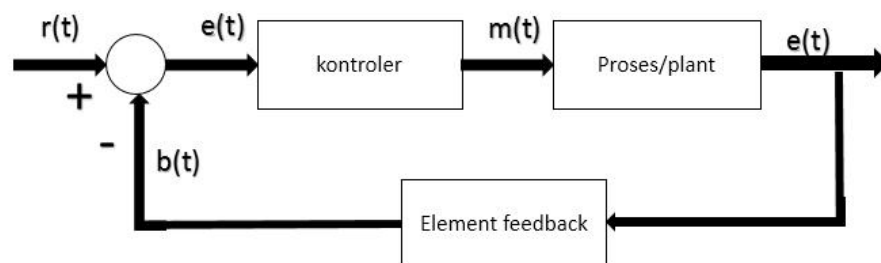
Perbandingan antara keluaran (output) terhadap masukan (input) suatu sistem pengendalian. Suatu misal fungsi alih sistem pengendalian loop terbuka Gambar 1 dapat dicari dengan membandingkan antara output terhadap input. Demikian pula fungsi alih pada Gambar 2.

8. Sistem Pengendalian Umpan Maju (*open loop system*)

Sistem kendali ini disebut juga sistem pengendalian lup terbuka . Pada sistem ini keluaran tidak ikut andil dalam aksi pengendalian.

9. Sistem Pengendalian Umpan Balik

Istilah ini sering disebut juga sistem pengendalian lup tertutup. Pengendalian jenis ini adalah suatu sistem pengaturan dimana sistem keluaran pengendalian ikut andil dalam aksi kendali. Gambar 3 menunjukkan blok diagram untuk sistem pengendalian lup tertutup.

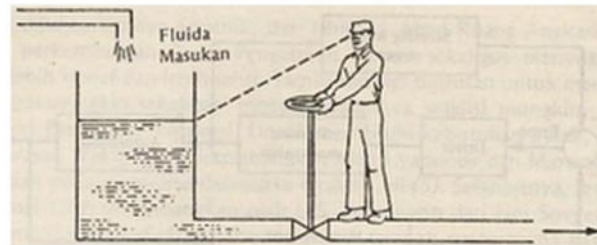


Gambar 3. Sistem Pengendalian lup tertutup
(Sumber : <http://www.eviandriani.com/2010/05/pengertian-sistem-kendali.html>)

10. Sistem Pengendalian Manual

Sistem pengendalian dimana faktor manusia sangat dominan dalam aksi pengendalian yang dilakukan pada sistem tersebut. Peran manusia sangat dominan dalam menjalankan perintah, sehingga hasil pengendalian akan dipengaruhi pelakunya. Pada sistem kendali manual ini juga termasuk dalam kategori sistem kendali jerat tertutup. Tangan berfungsi untuk mengatur permukaan fluida dalam tangki. Permukaan fluida dalam tangki bertindak sebagai masukan, sedangkan penglihatan bertindak sebagai sensor. Operator berperan membandingkan tinggi sesungguhnya saat itu dengan tinggi permukaan fluida yang dikehendaki, dan kemudian bertindak

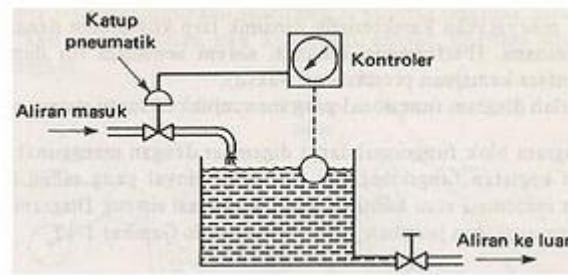
untuk membuka atau menutup katup sebagai aktuator guna mempertahankan keadaan permukaan yang diinginkan. Gambar 4 menunjukkan gambaran sistem pengendalian level cairan secara manual.



Gambar 4. Sistem pengendalian level cairan secara manual
(Sumber : <http://www.eviandriani.com/2010/05/pengertian-sistem-kendali.html>)

11. Sistem Pengendalian Otomatis

Sistem pengendalian dimana faktor manusia tidak dominan dalam aksi pengendalian yang dilakukan pada sistem tersebut. Peran manusia digantikan oleh sistem kontroler yang telah diprogram secara otomatis sesuai fungsinya, sehingga bisa memerankan seperti yang dilakukan manusia. Di dunia industri modern banyak sekali sistem kendali yang memanfaatkan kontrol otomatis, apalagi untuk industri yang bergerak pada bidang yang proses nya membahayakan keselamatan jiwa manusia. Gambar 5 menunjukkan gambaran sistem pengendalian level cairan secara otomatis.



Gambar 5. Sistem pengendalian level cairan secara otomatis
(Sumber : <http://www.eviandriani.com/2010/05/pengertian-sistem-kendali.html>)

12. Variabel terkendali (*ControlLED variable*)

Besaran atau variabel yang dikendalikan, biasanya besaran ini dalam diagram kotak disebut *process variable* (PV). Level fluida pada bejana pada Gambar 4 merupakan variabel terkendali dari proses pengendalian. Temperatur pada Gambar 5 merupakan contoh variabel terkendali dari suatu proses pengaturan.

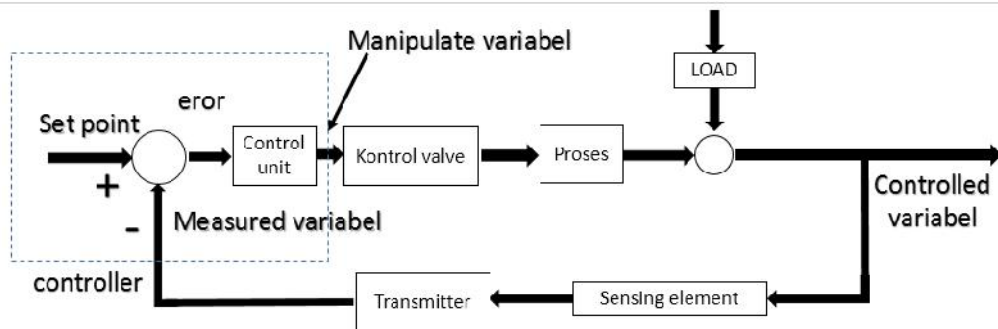
13. *Manipulated variable*

Masukan dari suatu proses yang dapat diubah-ubah atau dimanipulasi agar *process variable* besarnya sesuai dengan *set point* (sinyal yang diumpankan pada suatu sistem kendali yang digunakan sebagai acuan untuk menentukan keluaran sistem kontrol). Masukan proses pada Gambar 4 adalah laju aliran fluida yang keluar dari bejana, sedangkan masukan proses dari Gambar 5 adalah laju aliran fluida yang masuk menuju bejana. Laju aliran diatur dengan mengendalikan bukaan katup.

14. Sistem Pengendalian Digital

Dalam sistem pengendalian otomatis terdapat komponen-komponen utama seperti elemen proses, elemen pengukuran (*sensing element* dan

transmitter), elemen controller (*control unit*), dan *final control element* (*control value*). Gambar 6 menunjukkan blok diagram untuk sistem pengendalian digital.



Gambar 6. Sistem pengendalian digital
(Sumber : <http://www.eviandriani.com/2010/05/pengertian-sistem-kendali.html>)

15. Gangguan (*disturbance*)

Suatu sinyal yang mempunyai kecenderungan untuk memberikan efek yang melawan terhadap keluaran sistem pengendalian (variabel terkendali). Besaran ini juga lazim disebut *load*.

16. *Sensing element*

Bagian paling ujung suatu sistem pengukuran (*measuring system*) atau sering disebut sensor. Sensor bertugas mendeteksi gerakan atau fenomena lingkungan yang diperlukan sistem kontroler. Sistem dapat dibuat dari sistem yang paling sederhana seperti sensor on/off menggunakan limit switch, sistem analog, sistem bus paralel, sistem bus serial serta sistem mata kamera. Contoh sensor lainnya yaitu *thermocouple* untuk pengukur temperatur, *accelerometer* untuk pengukur getaran, dan *pressure gauge* untuk pengukur tekanan.

17. Transmitter

Alat yang berfungsi untuk membaca sinyal *sensing element* dan mengubahnya supaya dimengerti oleh kontroler.

18. Aktuator

Piranti elektromekanik yang berfungsi untuk menghasilkan daya gerakan. Perangkat bisa dibuat dari sistem motor listrik (motor DC servo, motor DC stepper, ultrasonik motor, *linier* motor, *torque* motor, solenoid), sistem pneumatik dan hidrolis. Untuk meningkatkan tenaga mekanik aktuator atau torsi gerakan maka bisa dipasang sistem gear box atau *sprocket chain*.

19. Transduser

Piranti yang berfungsi untuk mengubah satu bentuk energi menjadi energi bentuk lainnya atau unit pengalih sinyal. Suatu contoh mengubah sinyal gerakan mekanis menjadi energi listrik yang terjadi pada peristiwa pengukuran getaran. Terkadang antara transmitter dan transduser dirancukan, keduanya memang mempunyai fungsi serupa. Transduser lebih bersifat umum, namun transmitter penggunaannya pada sistem pengukuran.

20. *Measurement Variable*

Sinyal yang keluar dari transmitter, ini merupakan cerminan sinyal pengukuran.

21. *Setting point*

Besar variabel proses yang dikehendaki. Suatu kontroler akan selalu berusaha menyamakan variabel terkendali terhadap set point.

22. *Error*

Selisih antara *set point* dikurangi variabel terkendali. Nilainya bisa positif atau negatif, bergantung nilai *set point* dan variabel terkendali. Makin kecil *error* terhitung, maka makin kecil pula sinyal kendali kontroler terhadap *plant* hingga akhirnya mencapai kondisi tenang (*steady state*)

23. Alat Pengendali (*Controller*)

Alat pengendali sepenuhnya menggantikan peran manusia dalam mengendalikan suatu proses. *Controller* merupakan elemen yang mengerjakan tiga dari empat tahap pengaturan, yaitu:

- a. membandingkan *set point* dengan *measurement variable*
- b. menghitung berapa banyak koreksi yang harus dilakukan, dan
- c. mengeluarkan sinyal koreksi sesuai dengan hasil perhitungannya.

24. *Control Unit*

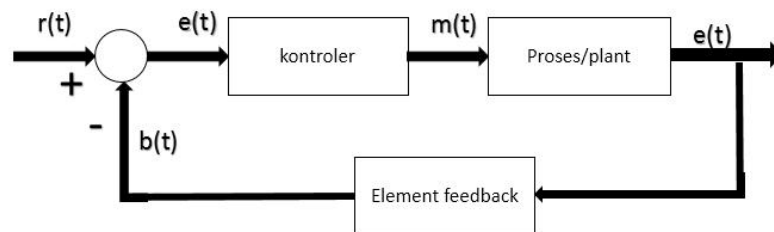
Bagian unit kontroler yang menghitung besarnya koreksi yang diperlukan.

25. *Final Controller Element*

Bagian yang berfungsi untuk mengubah *measurement variable* dengan memanipulasi besarnya *manipulated variable* atas dasar perintah kontroler.

26. Sistem Pengendalian Kontinyu

Sistem pengendalian yang berjalan secara kontinyu, pada setiap saat respon sistem selalu ada. Pada gambar 7. Sinyal $e(t)$ yang masuk ke kontroler dan sinyal $m(t)$ yang keluar dari kontroler adalah sinyal kontinyu.



Gambar 7. Sistem pengendalian kontinyu
(Sumber : <http://www.eviandriani.com/2010/05/pengertian-sistem-kendali.html>)

B. Robot

Kata “robot” diambil dari bahasa Ceko (Chech), yang memiliki arti “pekerja” (worker). Istilah ini muncul pada tahun 1920 oleh seorang pengarang bernama Karel Capek. Karyanya pada saat itu berjudul “*Rossum’s Universal Robot*”. Robot merupakan suatu perangkat mekanik yang mampu menjalankan tugas-tugas fisik, baik di bawah kendali dan pengawasan manusia (robot manual), ataupun yang dijalankan dengan serangkaian program yang telah didefinisikan terlebih dahulu (robot otomatis).

Robot modern terdiri dari tiga bagian penting yaitu sistem mekanik, elektronik dan program. Untuk menjalankan tugasnya robot mengambil besaran atau data dari sensor yang diproses menggunakan prosesor atau kontroler kemudian menggerakkan aktuator sesuai perintah dalam prosesor.

C. Bulutangkis

Bulutangkis adalah suatu permainan yang menggunakan sebuah raket dan shuttlecock yang di pukul melewati sebuah net. permainan ini berlaku untuk siapa saja dengan bentuk tunggal (*single*), juga dengan ganda (*double*), dan dengan ganda campuran (*mixed double*).

Permainan ini biasanya dimainkan oleh:

1. seorang pria melawan seorang pria (tunggal pria atau *men's single*),
2. seorang wanita melawan seorang wanita (tunggal wanita atau *women's single*),
3. sepasang pria melawan sepasang pria (ganda pria atau *men's double*),
4. sepasang wanita melawan sepasang wanita (ganda putrid atau *women's double*),
5. sepasang pria dan wanita melawan sepasang pria dan wanita (ganda campuran atau *mixed doubles*).

Tiap partai terdiri dari maksimal tiga set (*the best of three games*), yang di sebut set kesatu, kedua, dan ketiga. Jika permainan menang dua set berturut-turut disebut menang dengan langsung (*straight-set*). Sedangkan jika menang dengan satu lawan satu, disebut dengan menang set panjang (*rubber-set*).

Tiap set terdiri dari 15 angka, kecuali untuk tunggal wanita hana 11 angka, jika pemain atau pasangan pemain yang telah mencapai angka-angka tersebut itu memenangkan set tersebut. Olahraga bulutangkis dimainkan di atas lapangan yang di batasi dengan garis-garis dalam ukuran panjang dan lebar tertentu. Lapangan di bagi dua sama besar dan di pisahkan oleh net yang diregangkan di kedua tiang

net yang ditanam di pinggir lapangan. Alat yang di pergunakan adalah raket sebagai alat pemukul, serta shuttlecocks sebagai bola yang dipukul.

D. *Wireless*

Wireless merupakan Koneksi antar suatu perangkat dengan perangkat lainnya tanpa menggunakan kabel atau Metode untuk mengirimkan sinyal melalui suatu ruangan bukannya menggunakan kabel. Gelombang radio dan sinar infra merah biasa digunakan untuk komunikasi nirkabel.

1. *Wireless Application Protocol*

Disingkat dengan WAP. Standar protokol untuk aplikasi *wireless* (seperti yang digunakan pada ponsel). WAP adalah sebuah protocol atau sebuah teknik *messaging service* yang memungkinkan sebuah hp digital atau terminal *mobile* yang mempunyai fasilitas WAP, melihat/membaca isi sebuah situs di internet dalam sebuah format text khusus. Situs internet ini harus merupakan situs dengan fasilitas WAP.

Teknologi ini merupakan hasil kerjasama antar industri untuk membuat sebuah standar yang terbuka dan berbasis pada standar Internet, serta beberapa protokol yang sudah dioptimasi untuk lingkungan *wireless*.

Teknologi ini bekerja dalam modus teks dengan kecepatan sekitar 9,6 kbps. Belakangan juga dikembangkan protokol GPRS yang memiliki beberapa kelebihan dibandingkan WAP.

Wireless Application Protocol merupakan sebuah protocol pengembangan dari protocol *wireless* data yang telah ada. Phone.com menciptakan sebuah versi standart HTML (*HyperText Markup Language*) Internet *protocol* yang didesain

khusus untuk transfer informasi antar *mobile network* yang efisien. Terminal *wireless* dengan HDML (*Handheld Device Markup Language*) *microbrowser*, dan *Handheld Device Transport Protocol* (HDTP) dari Phone.com terhubung dengan UP. *Link Server Suite* yang seterusnya terhubung ke Internet atau intranet dimana informasi yang dibutuhkan berada. Teknologi inilah yang kemudian dikenal sebagai WAP.

Keterbatasan perangkat *Wireless Application Protocol* antara lain:

- a. kemampuan *Central Processing Unit* (CPU) yang lebih rendah dibandingkan CPU yang digunakan pada perangkat *wired* (seperti komputer),
- b. keterbatasan ukuran memori.
- c. penghematan penggunaan catu daya yang biasanya menggunakan baterai,
- d. ukuran tampilan yang lebih kecil dan terbatas,
- e. perangkat masukan yang berbeda dengan perangkat biasa.

Desain dari informasi yang dikirimkan melalui WAP biasanya menggunakan format WML, *Wireless Markup Language*. WML ini mirip HTML, hanya lebih spesifik untuk perangkat *wireless* yang memiliki keterbatasan seperti di atas.

2. *Wireless Bitmap*

Wireless Bitmap isingkat dengan WBMP. Format grafik yang terdapat dalam WAP. WBMP merupakan format yang mirip dengan format BMP. Gambar dengan standar format WBMP terbagi dalam dua bagian sebagai berikut.

- a. Bagian *header*, tempat untuk informasi karakteristik dari gambar, seperti tinggi, lebar dan tipe gambar.

- b. Bagian isi yang disebut *Type dependent*, merupakan bagian dari informasi gambar.

Standar format WBMP ini dibuat dengan susunan yang dapat diperluas kegunaannya. Bagian isi dapat berubah menjadi format-format baru yang dapat diakses. Walaupun WBMP akan memperlambat transfer data karena ukurannya yang tidak kecil, tetapi menggunakan gambar dapat memberikan informasi yang lebih banyak pada layar berukuran kecil seperti pada handphone.

3. *Wireless Computing*

Wireless Computing merupakan proses komputerisasi yang dilakukan melalui media jaringan tanpa kabel.

4. *Wireless Fidelity*

Wireless Fidelity biasa disingkat dengan WiFi. WiFi merupakan sebuah teknologi yang memungkinkan sejumlah komputer terhubung dalam sebuah jaringan tanpa kabel alias *wireless LAN*.

5. *Wireless Internet Service Provider*

Disingkat dengan WISP. *Internet Access Provider* atau *Internet Service Provider* yang berusaha memberikan layanan sambungan nirkabel *broadband* dan sambungan untuk station bergerak kepada perusahaan pengguna.

6. *Wireless Markup Language*

Disingkat dengan WML. *Wireless Markup Language* merupakan Sebuah standar bahasa yang mirip HTML hanya dikhususkan kepada perangkat *wireless* seperti *handphone*.

7. *Wireless Entertainment Device*

Wireless Entertainment Device adalah PDA yang berukuran saku yang memiliki kemampuan konektivitas nirkabel dan dirancang untuk online gaming, konsepnya ialah penggabungan dari telepon seluler dan perangkat seperti *Gameboy*.

8. *Wireless Card*

Wireless Card merupakan kartu yang digunakan untuk mendukung komputer bisa terhubung dalam suatu jaringan. Kartu ini biasanya digunakan pada notebook yang disebut dengan PCMCIA (*Personal Computer Memory Card International Association*).

9. *Wireless LAN*

Wireless LAN adalah jaringan komputer yang terhubung melalui tanpa kabel. Local Area Network dari komputer dan peralatan lainnya yang berkomunikasi lewat sinyal radio atau gelombang cahaya. Sistem ini berguna apabila penyambungan lewat koneksi kabel atau serat optik cukup mahal atau untuk aplikasi koneksi bergerak.

Teknologi komunikasi data dengan tidak menggunakan kabel untuk menghubungkan antara klien dan server. Secara umum teknologi *Wireless LAN* hampir sama dengan teknologi jaringan komputer yang menggunakan. Teknologi *Wireless LAN* ada yang menggunakan frekuensi radio untuk mengirim dan menerima data yang tentunya mengurangi kebutuhan atau ketergantungan hubungan melalui kabel. Akibatnya pengguna mempunyai mobilitas atau fleksibilitas yang tinggi dan tidak tergantung pada suatu tempat atau lokasi. Teknologi *Wireless LAN* juga memungkinkan untuk membentuk jaringan

komputer yang mungkin tidak dapat dijangkau oleh jaringan komputer yang menggunakan kabel.

10. *Wireless* PAN

Personal Area Network yang terhubung dengan media tanpa kabel. Teknologi yang digunakan pada *wireless* PAN ini adalah IrDA dan Bluetooth.

11. *Wireless* Modem

Wireless Modem merupakan modem yang digunakan untuk jaringan tanpa kabel.

12. Infra Merah

Gelombang cahaya infra merah. Gelombang ini dapat digunakan untuk proses transmisi data untuk jarak dekat.

Standard *wireless networking* yang diluncurkan pada dasarnya adalah menggunakan hubungan radio jarak dekat atau *short-range radio link* untuk pertukaran informasi, sehingga hubungan antar hp, mobile PC, PDA, dan lainnya dapat dilakukan tanpa gangguan kabel atau *wireless*.

13. Bluetooth

Tujuan dari peluncuran bluetooth ini diantaranya adalah untuk mengganti spesifikasi IrDA dari Infra merah pada hp dan peralatan *mobile* lainnya. *Bluetooth* menyediakan transfer data 720 Kbps dalam range 40 feet. *Bluetooth* menggunakan gelombang radio yang omni direksional dan dapat menembus dinding. Ini berbeda dengan IrDa yang menggunakan teknologi pandang dan perlu satu sama lain agar bisa melakukan kontak.

Ericsson memberikan sumbangan mereka pada teknologi radio, Toshiba dan IBM mengembangkan spesifikasi untuk mengintegrasikan teknologi *Bluetooth* kedalam peralatan *mobile*. Intel menyumbangkan keahlian mereka dalam chip dan software sedangkan Nokia menyumbangkan keahlian mereka dalam teknologi radio dan *mobile handset software*.

Banyak perusahaan lain juga diundang untuk mendukung teknologi ini sehingga diharapkan teknologi ini dapat dipakai dalam banyak peralatan. Radio ini akan beroperasi pada 2.45 GHz ISM (*Industrial Scientific Medical*), yang memungkinkan pengguna internasional dengan peralatan yang dilengkapi dengan Bluetooth dapat menggunakan peralatan mereka dimana saja di seluruh dunia.

Nama *Bluetooth* berasal dari King Harald Bluetooth dari Denmark. Ericsson (suatu perusahaan Skandinavia) adalah perusahaan yang pertama kali mengembangkan spesifikasi ini.

14. 3G

3G atau *third generation* adalah istilah yang digunakan untuk sistem komunikasi *mobile* generasi selanjutnya. Sistem ini akan memberikan pelayanan yang lebih baik dari apa yang ada sekarang, yaitu pelayanan suara, text dan data.

Jasa layanan yang diberikan oleh 3G ini adalah Jasa pelayanan Video, akses ke multimedia dan *mobile Internet* kecepatan tinggi, adalah beberapa kemungkinan yang akan didapat oleh konsumen pada masa yang akan datang. Sistem 3rd *Generation* akan memperbesar kemungkinan pada sistem komunikasi dan informasi.

Keuntungan utama adalah sistem ini akan menawarkan pelayanan dengan kapabilitas *high-end*, yang mana termasuk peningkatan kapasitas, kualitas dan data *rate* dari apa yang ada sekarang. Juga akan dapat melakukan pemakaian serentak dari beberapa jasa pelayanan. Sistem 3rd *Generation* juga akan menjembatani celah yang ada antara dunia *wireless* dan dunia computer/internet.

15. *Service Set Identifier*

Nama dari suatu *wireless local area network*, digunakan pada semua perangkat nirkabel agar bisa untuk berkomunikasi satu sama lainnya.

16. Antena

Antena merupakan suatu alat yang digunakan untuk mengirim maupun untuk menerima suatu sinyal. Antena ini lebih ditujukan untuk jaringan tanpa kabel, seperti antena televisi, antena handphone, antena untuk WLAN (*Wireless Local Area Network*), *satellite dish*.

E. Mikrokontroler AVR (Alf and Vegaard's Risc Processor)

Mikrokontroler adalah prosesor yang sekarang sudah banyak dikenal dan digunakan secara luas pada dunia industri. Mikrokontroler saat ini merupakan *Chip* utama pada hampir setiap peralatan elektronika canggih. Robot-robot canggih pun bergantung pada kemampuan mikrokontroler dan ketekunan pembuat program mikrokontroler tersebut. Keuntungan menggunakan mikrokontroler yaitu selain harganya murah, banyak ditemui di pasaran, dapat diprogram berulang-ulang dan dapat diprogram sesuai keinginan.

Mikrokontroler jenis AVR adalah prosesor yang sekarang ini paling banyak digunakan dalam membuat aplikasi sistem kendali bidang instrumentasi,

dibandingkan dengan mikrokontroler keluarga MCS51 seperti AT 89C51/52. AVR adalah mikrokontroler RISC (*Reduce Instruction Set Compute*) 8 bit berdasarkan arsitektur *Hardvard*, yang dibuat oleh Atmel pada tahun 1996. AVR mempunyai kepanjangan *Advanced Versatile RISC* atau *Alf and Vagard's Risc processor* yang berasal dari dua nama mahasiswa *Noewegian Institute of Technology* (NHT), yaitu Alf-Egil Bogen dan Vegard Wollan.

Mikrokontroler seri AVR pertama kali diperkenalkan ke pasaran sekitar tahun 1997 oleh perusahaan Atmel, yaitu sebuah perusahaan yang sangat terkenal dengan produk mikrokontroler seri AT89S51/52-nya yang sampai sekarang masih banyak digunakan di lapangan. Keterbatasan pada mikrokontroler tersebut (resolusi, memori, dan kecepatan) menyebabkan banyak orang beralih ke mikrokontroler AVR. Hal ini karena ada beberapa kelebihan dari tipe AVR ini yaitu diantaranya *ADC internal*, *EEPROM internal*, *Counter/Timer*, *I2C*, *USART*, *PWM*, *Port I/O*, komunikasi *serial*, dan sebagainya.

Mikrokontroler AVR standar memiliki arsitektur 8 bit, dimana semua instruksi dikemas dalam kode 16 bit dan sebagian besar instruksi dieksekusi dalam satu siklus *clock*, berbeda dengan instruksi MCS51 yang membutuhkan 12 siklus *clock* (Widodo Budiharto dan Gamayel Rizal, 2007:28). Hal ini karena kedua jenis mikrokontroler tersebut memiliki arsitektur yang berbeda. AVR berteknologi RISC(*Reduce Insruction Set Computing*), sedangkan seri MCS51 berteknologi CISC(*Complex Instruction Set Computing*). AVR dapat dikelompokkan menjadi empat kelas yaitu keluarga ATtiny, keluarga AT90Sxx, keluarga ATmega, dan

AT86RFFxx. Perbedaan dari masing - masing keluarga AVR tersebut adalah memori, peripheral, dan fungsinya.

F. Mikrokontroler ATmega16

Mikrokontroler ATmega16 adalah mikrokontroler yang diproduksi oleh Atmel. Mikrokontroler ini memiliki clock dan kerjanya tinggi sampai 16 MHz, ukuran flash memorinya cukup besar, kapasistas SRAM sebesar 2 KiloByte, dan 32 buah port I/O.

1. Arsitektur CPU ATmega16

Fungsi utama CPU adalah memastikan pengeksekusian instruksi dilakukan dengan benar. Oleh karena itu CPU harus dapat mengakses memori, melakukan kalkulasi, mengontrol peripheral, dan menangani interupsi. Ada 32 buah *General Purpose Register* yang membantu ALU bekerja. Untuk operasi aritmatika dan logika, *operand* berasal dari dua buah *general register* dan hasil operasi ditulis kembali ke *register*. Status and *control* berfungsi untuk menyimpan instruksi aritmatika yang baru saja dieksekusi. Informasi ini berguna untuk mengubah alur program saat mengeksekusi operasi kondisional. Instruksi di jemput dari flash memory. Setiap byte flash memory memiliki alamat masing-masing.

Alamat instruksi yang akan dieksekusi senantiasa disimpan program counter. Ketika terjadi interupsi atau pemanggilan rutin biasa, alamat di program counter disimpan terlebih dahulu di *stack*. Alamat interupsi atau rutin kemudian ditulis ke program counter, instruksi kemudian dijemput dan dieksekusi. Ketika CPU telah selesai mengeksekusi rutin interupsi atau rutin biasa, alamat yang ada di stack dibaca dan ditulis kembali ke program counter.

2. Program memori

ATmega16 memiliki 16 KiloByte flash memori untuk menyimpan program. Karena lebar intruksi 16 bit atau 32 bit maka flash memori dibuat berukuran 16Kx16, artinya ada 16K alamat di flash memori yang bisa dipakai dimulai dari alamat 0 heksa sampai alamat 3FFF heksa dan setiap alamatnya menyimpan 16 bit instruksi.

3. SRAM data memori

ATmega16 memiliki 2 KiloByte SRAM. Memori ini dipakai untuk menyimpan variabel. Tempat khusus di SRAM yang senantiasa ditunjuk register SP disebut *stack*. *Stack* berfungsi untuk menyimpan nilai yang *dipush*.

4. EEPROM data memori

ATmega16 memiliki 1024 byte data EEPROM. Data di EEPROM tidak akan hilang walaupun catuan daya ke sistem mati. Parameter sistem yang penting disimpan di EEPROM. Saat sistem pertama kali menyala parameter tersebut dibaca dan system diinisialisasi sesuai dengan nilai parameter tersebut.

5. Interupsi

Sumber interupsi ATmega16 ada 21 buah. Saat interupsi diaktifkan dan interupsi terjadi maka CPU menunda instruksi sekarang dan melompat ke alamat rutin interupsi yang terjadi. Setelah selesai mengeksekusi intruksi-instruksi yang ada di alamat rutin interupsi CPU kembali melanjutkan instruksi yang sempat tertunda.

6. I/O port

ATmega16 memiliki 32 buah pin I/O. Melalui pin I/O inilah ATmega16 berinteraksi dengan sistem lain. Masing-masing pin I/O dapat dikonfigurasi tanpa mempengaruhi fungsi pin I/O yang lain. Setiap pin I/O memiliki tiga register yakni: DDxn, PORTxn, dan PINxn. Kombinasi nilai DDxn dan PORTxn menentukan arah pin I/O.

7. *Clear timer on compare match* (CTC)

CTC adalah salah satu mode *Timer/Counter1*, selain itu ada Normal mode, *FastPWM mode*, *Phase Correct PWM mode*. Pada CTC mode maka nilai TCNT1 menjadi nol jika nilai TCNT1 telah sama dengan OCR1A atau ICR1. Jika nilai top ditentukan OCR1A dan interupsi diaktifkan untuk *Compare Match A* maka saat nilai TCNT1 sama dengan nilai OCR1A interupsi terjadi. CPU melayani interupsi ini dan nilai TCNT1 menjadi nol.

8. USART

Selain untuk general I/O, pin PD1 dan PD0 ATmega16 berfungsi untuk mengirim dan menerima bit secara serial. Pengubahan fungsi ini dibuat dengan mengubah nilai beberapa register serial. Untuk menekankan fungsi ini, pin PD1 disebut TxD dan pin PD0 disebut RxD.

G. Fitur Mikrokontroler ATmega16

Fitur-fitur yang terdapat pada mikrokontroler ATmega16 antara lain adalah sebagai berikut.

1. *High-Performance, Low-Power AVR 8-bit RISC microcontroller*
2. *Advanced RISC Architecture*

- a. *131 Powerful Instructions – Most Single-clock Cycle Execution*
- b. *32 x 8 General Purpose Working Registers*
- c. *Fully Static Operation*
- d. *Up to 16 MIPS Throughput at 16MHz*
- e. *On-chip 2-cycle Multiplier*
- 3. *High-Endurance Non-Volatile Memory segments*
 - a. *16K Bytes of In-System Self-programmable Flash*
Endurance: 10,000 Write/Erase Cycles
 - b. *Optional Boot Code Section with Independent Lock Bits*
In-System Programming by On-chip Boot Program
True Read-While-Write Operation
 - c. *1024Bytes EEPROM*
Endurance: 100,000 Write/Erase Cycles
 - d. *2K Byte Internal SRAM*
 - e. *Programming Lock for Software Security*
- 4. *JTAG (IEEE std. 1149.1 Compliant) Interface*
 - a. *Boundary-scan Capabilities According to the JTAG Standard*
 - b. *Extensive On-chip Debug Support*
 - c. *Programming of Flash, EEPROM, Fuses and Lock Bits through the JTAG Interface*
- 5. *Peripheral features*
 - a. *Two 8-bit Timers/Counters with Separate Prescaler and Compare Mode*

- b. One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode*
- c. Real Time Counter with Separate Oscillator*
- d. Four PWM Channels*
- e. 8-channel, 10-bit ADC*
- f. 8 Single-ended Channels*
- g. 7 Differential Channels in TQFP Package Only 2 Differential Channels with Programmable Gain at 1x, 10x, 200x, Byte-oriented Two-wire Serial Interface*
- h. Programmable Serial USART, Master/Slave SPI Serial Interface*
- i. Programmable Watchdog Timer with Separate On-Chip Oscillator*
- j. On-Chip Analog Comparator*
- 6. Special Microcontroller features*
 - a. Power-On Reset and Programmable Brown-out Detection*
 - b. Internal Calibrated RC Oscillator*
 - c. External and Internal Interrupt Sources*
 - d. Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby*
- 7. I/O and Packages*
 - a. 32 Programmable I/O Lines*
 - b. 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF*
- 8. Operating Voltages*
 - a. 2.7 – 5.5V (ATmega16L)*
 - b. 4.5 – 5.5V (ATmega16)*

9. *Speed Grades*

a. *0 – 8MHz (ATmega16L) , 0 – 16MHz (ATmega16)*

10. *Power Consumption at 1MHz, 3V, 25`C for ATmega16L*

a. *Active: 1.1 mA*

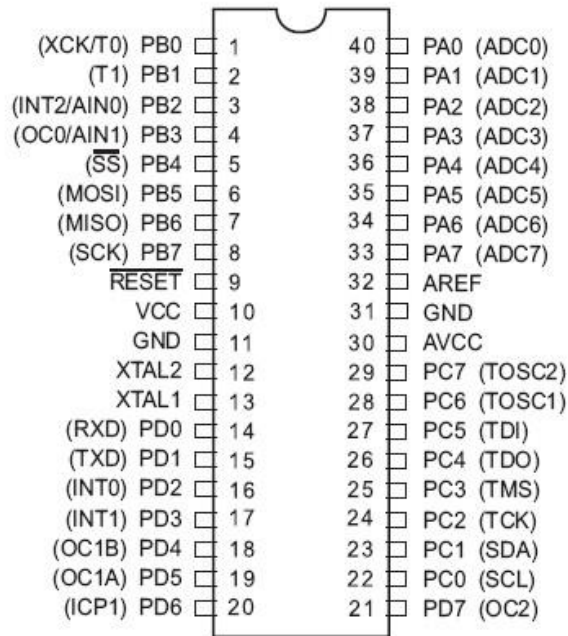
b. *Idle Mode: 0.35 mA*

c. *Power-Down Mode: <1 uA*

Gambar 8. Blok diagram fungsional ATmega16
(Sumber : Data Sheet ATmega16)

I. Konfigurasi Pin AVR ATmega16

Konfigurasi pin mikrokontroler ATmega16 dapat dilihat pada Gambar 9.



Gambar 9. Susunan Pin ATmega16
(Sumber : Data Sheet ATmega16)

Berikut ini adalah susunan pin/kaki dari ATmega16.

1. VCC sebagai masukan catu daya.
2. GND
3. Port A (PA.0-7) sebagai *input/output* dua arah dan masukan ADC.
4. Port B (PB.0-7) *input/output* dua arah dan pin fungsi khusus seperti SCK, MISO, MOSI, SS, OC0/AIN1, INT2/AIN0, T1, XCK/T0.
5. Port C (PC.0-7) *input/output* dua arah dan pin fungsi khusus seperti TOSC2, TOSC1, TDI, TDO, TMS, TCK, SDA, SCL
6. Port D (PD.0-7) *input/output* dua arah dan pin fungsi khusus seperti RXD, TXD, INT0, INT1, OC1B, OC1A, ICP1, OC2
7. *Reset* untuk mereset mikrokontroler.

8. XTAL 1 dan 2 merupakan pin *clock eksternal*.
9. AVCC untuk tegangan masukan fungsi *ADC*.
10. AREF untuk tegangan referensi *eksternal ADC*.

J. Status Register

Status register adalah register yang berisi status yang dihasilkan pada setiap operasi yang dilakukan ketika suatu instruksi dijalankan (dieksekusi). SREG merupakan bagian dari CPU mikrokontroler. Berikut ini adalah status register dari ATmega16 beserta penjelasannya. Urutan status register ATmega16 dapat dilihat pada Gambar 10.

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 10. Status register ATmega16
(Sumber : Data Sheet ATmega16)

1. Bit 7 (I)

Merupakan bit *Global Interrupt Enable*. Bit ini harus di-set supaya semua perintah interupsi dapat dijalankan. Untuk fungsi interupsi individual akan dijelaskan pada bagian lain. Jika bit ini di-set, maka semua perintah interupsi baik yang individual maupun secara umum akan diabaikan. Bit ini akan dibersihkan atau *cleared* oleh *hardware* setelah sebuah interupsi dijalankan dan akan di-set kembali oleh perintah RETI. Bit ini juga dapat di-set dan di-reset melalui aplikasi dengan instruksi SEI dan CLI.

2. BIT 6 (T)

Merupakan bit *Copy Storage*. Instruksi bit *Copy Instructions* BLD (*Bit Load*) dan BST (*Bit Store*) menggunakan bit ini sebagai asal atau tujuan untuk bit yang telah dioperasikan. Sebuah bit dari sebuah register dalam *Register File* dapat disalin ke dalam bit ini dengan menggunakan instruksi BST, dan sebuah bit di dalam bit ini dapat disalin ke dalam sebuah bit di dalam register pada *Register File* dengan menggunakan perintah BLD.

3. BIT 5 (H)

Merupakan bit *Half Carry Flag*. Bit ini menandakan sebuah *Half Carry* dalam beberapa operasi aritmatika. Bit ini berfungsi dalam aritmatik BCD.

4. BIT 4 (S)

Merupakan *Sign* bit. Bit ini selalu merupakan sebuah eksklusif diantara *Negative Flag(N)* dan *Two's Complement Overflow Flag(V)*.

5. BIT 3 (V)

Merupakan bit *Two's Complement Overflow Flag*. Bit ini menyediakan fungsi-fungsi aritmatika dua komplemen.

6. BIT 2 (N)

Merupakan bit *Negative Flag*. Bit ini mengindikasikan sebuah hasil negatif di dalam sebuah fungsi logika atau aritmatika.

7. BIT 1 (Z)

Merupakan bit *Zero Flag*. Bit ini mengindikasikan sebuah hasil nol "0" dalam sebuah fungsi aritmatika atau logika.

8. BIT 0 (C)

Merupakan bit *Carry Flag*. Bit ini mengindikasikan sebuah *cary* atau sisa dalam sebuah fungsi aritmatika atau logika.

K. Mikrokontroler ATmega128

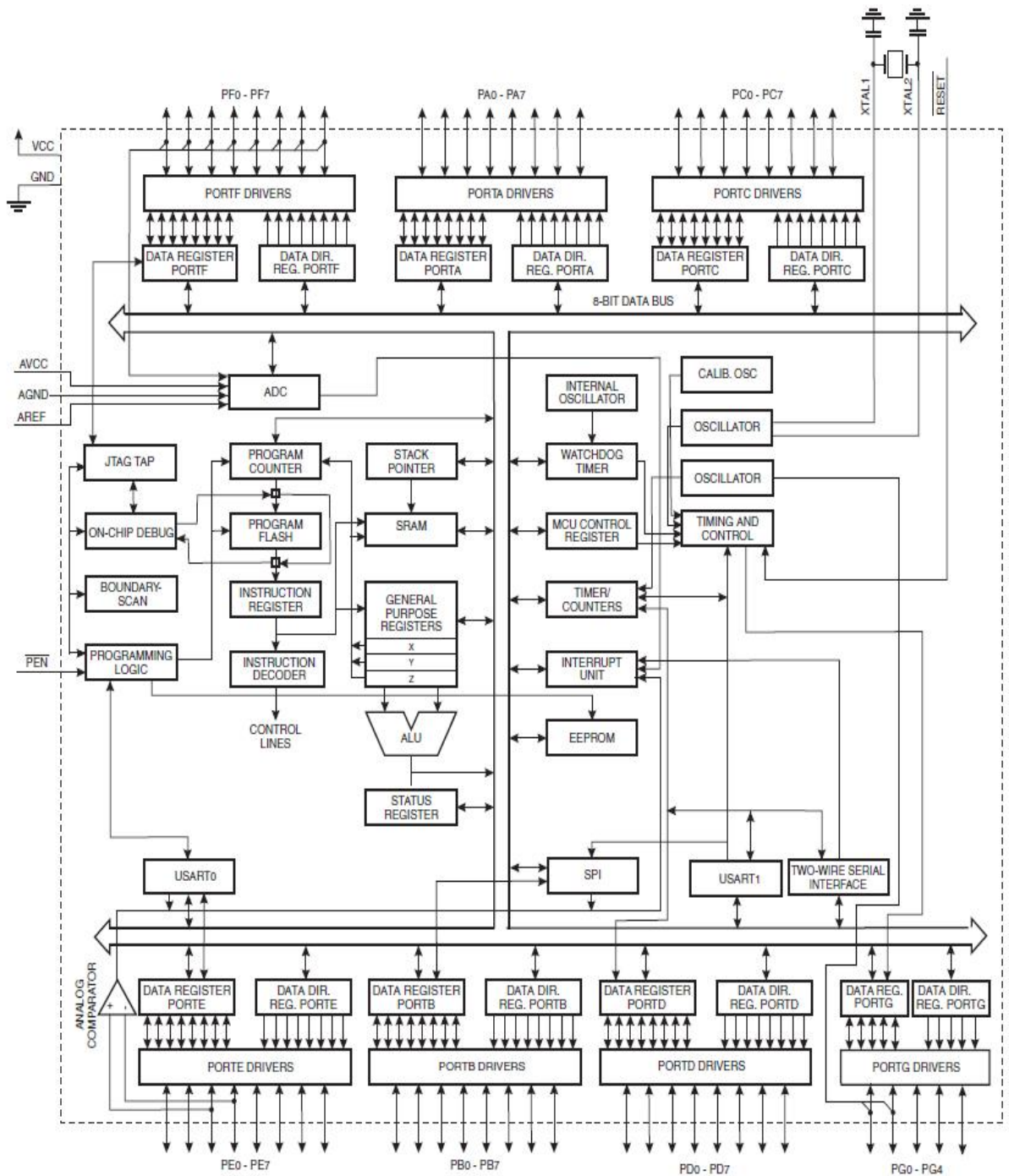
Mikrokontroller ATmega128 merupakan mikrokontroller keluarga AVR yang mempunyai kapasitas flash memori 128KB. AVR (*Alf and Vegard's Risc Processor*) merupakan seri mikrokontroller CMOS 8-bit buatan Atmel, berbasis arsitektur RISC (*Reduced Instruction Set Computer*).

1. Spesifikasi Mikrokontroller AVR ATmega128

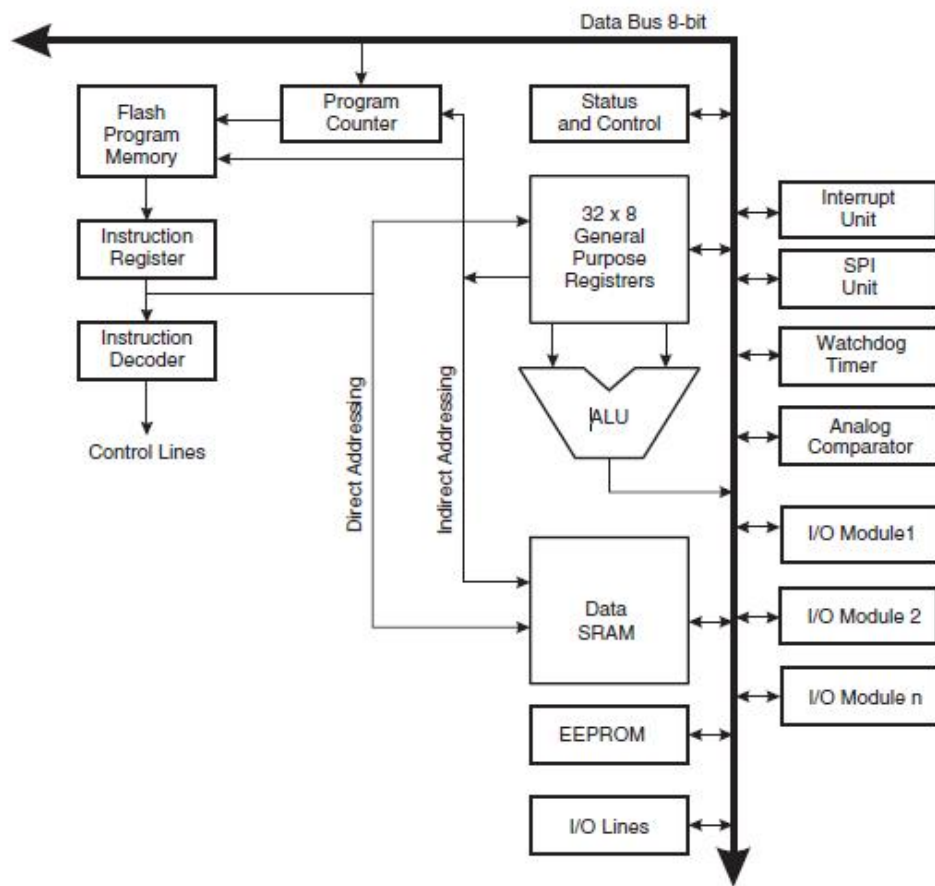
AVR ATmega128 memiliki spesifikasi sebagai berikut.

- a. Saluran I/O sebanyak 53 buah, yaitu Port A, Port B, Port C, Port D, Port E, Port F dan Port G.
- b. ADC 10 bit sebanyak 8 saluran.
- c. 2 buah *Timer/Counter* 8 bit dan 2 buah *Timer/Counter* 16 bit.
- d. Dua buah PWM 8 bit.
- e. *Watchdog Timer* dengan osilator internal.
- f. Internal SRAM sebesar 4 KByte.

Arsitektur Mikrokontroler ATmega128 ATmega16 dapat dilihat pada Gambar 11 dan Blok Diagram Fungsional ATmega128 dapat dilihat pada Gambar 12.



Gambar 11. Arsitektur Mikrokontroler ATmega128
(Sumber: Data Sheet ATmega128)

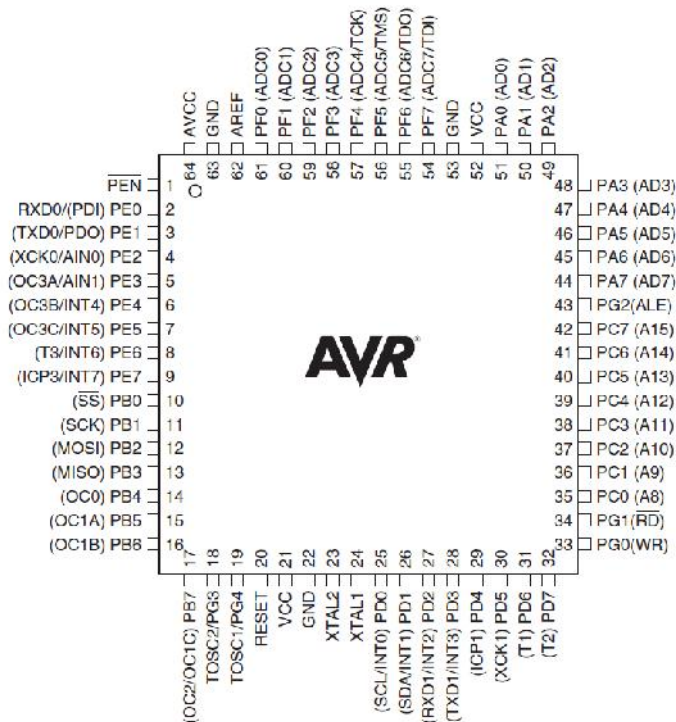


Gambar 12. Blok Diagram Fungsional ATmega128
(Sumber: Data Sheet ATmega128)

- g. Memori *flash* sebesar 128 KBytes.
- h. Interupsi Eksternal.
- i. Port antarmuka SPI.
- j. EEPROM sebesar 4 kbyte.
- k. *Real time counter*.
- l. 2 buah Port USART untuk komunikasi serial.
- m. Enam kanal PWM.
- n. Tegangan operasi sekitar 4,5 V sampai dengan 5,5V.

2. Konfigurasi Pin Mikrokontroler

Konfigurasi pin mikrokontroler ATmega128 bisa dilihat pada Gambar 13 berikut.



Gambar 13. Konfigurasi Pin ATmega128
(Sumber: Data Sheet ATmega128)

Dari Gambar 13 tersebut dapat dijelaskan secara fungsional konfigurasi pin ATmega128 sebagai berikut.

- VCC merupakan pin yang berfungsi sebagai pin masukan catu daya.
- GND merupakan pin ground.
- Port A (PA0...PA7)

Fungsi khusus port-port pada ATmega 128 dapat dilihat pada Tabel 1-7.

Tabel 1. Fungsi Khusus Port A ATmega128

Port Pin	Fungsi Alternatif
PA7	AD7 (External memory interface address and data bit 7)
PA6	AD6 (External memory interface address and data bit 6)
PA5	AD5 (External memory interface address and data bit 5)
PA4	AD4 (External memory interface address and data bit 4)
PA3	AD3 (External memory interface address and data bit 3)
PA2	AD2 (External memory interface address and data bit 2)
PA1	AD1 (External memory interface address and data bit 1)
PA0	AD0 (External memory interface address and data bit 0)

(Sumber: Data Sheet ATmega128)

d. Port B (PB0...PB7)

Tabel 2. Fungsi Khusus Port B ATmega128

Port Pin	Fungsi Alternatif
PB7	OC2/OC1C (Output Compare and PWM Output for Timer/Counter2 or Output Compare and PWM Output C for Timer/Counter1)
PB6	OC1B (Output Compare and PWM Output B for Timer/Counter1)
PB5	OC1A (Output Compare and PWM Output A for Timer/Counter1)
PB4	OC0 (Output Compare and PWM Output for Timer/Counter0)
PB3	MISO (SPI Bus Master Input/Slave Output)
PB2	MOSI (SPI Bus Master Output/Slave Input)
PB1	SCK (SPI Bus Serial Clock)
PB0	SS (SPI Slave Select input)

(Sumber: Data Sheet ATmega128)

e. Port C (PC0...PC7) merupakan pin I/O dua arah dan pin fungsi khusus
(*External Memory Interface*)

Tabel 3. Fungsi Khusus Port C ATmega128

Port Pin	Fungsi Alternatif
PC7	A15
PC6	A14
PC5	A13
PC4	A12
PC3	A11
PC2	A10
PC1	A9
PC0	A8

(Sumber: Data Sheet ATmega128)

f. Port D (PD0...PD8)

Tabel 4. Fungsi Khusus Port D ATmega128

Port Pin	Fungsi Alternatif
PD7	T2 (Timer/Counter2 Clock Input)
PD6	T1 (Timer/Counter1 Clock Input)
PD5	XCK1 (USART1 External Clock Input/Output)
PD4	ICP1 (Timer/Counter1 Input Capture Pin)
PD3	INT3/TXD1 (External Interrupt3 Input or UART1 Transmit Pin)
PD2	INT2/RXD1 (External Interrupt2 Input or UART1 Receive Pin)
PD1	INT1/SDA (External Interrupt1 Input or TWI Serial Data)
PD0	INT0/SCL (External Interrupt0 Input or TWI Serial Clock)

(Sumber: Data Sheet ATmega128)

g. Port E (PE0...PE8) merupakan pin I/O dua arah dan fungsi khusus yang dapat dilihat pada Tabel 5.

Tabel 5. Fungsi Khusus Port E ATmega128

Port Pin	Fungsi Alternatif
PE7	INT7/ICP3 (External Interrupt 7 Input or Timer/Counter3 Input Capture Pin)
PE6	INT6/ T3 (External Interrupt 6 Input or Timer/Counter3 Clock Input)
PE5	INT5/OC3C (External Interrupt 5 Input or Output Compare and PWM Output C for Timer/Counter3)
PE4	INT4/OC3B (External Interrupt4 Input or Output Compare and PWM Output B for Timer/Counter3)
PE3	AIN1/OC3A (Analog Comparator Negative Input or Output Compare and PWM Output A for Timer/Counter3)

Lanjutan Tabel 5.

Port Pin	Fungsi Alternatif
PE2	AIN0/XCK0 (Analog Comparator Positive Input or USART0 external clock input/output)
PE1	PDO/TXD0 (Programming Data Output or UART0 Transmit Pin)
PE0	PDI/RXD0 (Programming Data Input or UART0 Receive Pin)

(Sumber: Data Sheet ATmega128)

h. Port F (PF0...PF8) merupakan pin I/O dua arah dan fungsi khusus yang dapat dilihat pada Tabel 6.

Tabel 6. Fungsi Khusus Port F ATmega128

Port Pin	Fungsi Alternatif
PF7	ADC7/TDI (ADC input channel 7 or JTAG Test Data Input)
PF6	ADC6/TDO (ADC input channel 6 or JTAG Test Data Output)
PF5	ADC5/TMS (ADC input channel 5 or JTAG Test Mode Select)
PF4	ADC4/TCK (ADC input channel 4 or JTAG Test Clock)
PF3	ADC3 (ADC input channel 3)
PF2	ADC2 (ADC input channel 2)
PF1	ADC1 (ADC input channel 1)
PF0	ADC0 (ADC input channel 0)

(Sumber: Data Sheet ATmega128)

i. Port G (PG0...PG4) merupakan pin I/O dua arah dan fungsi khusus yang dapat dilihat pada Tabel 7.

Tabel 7. Fungsi Khusus Port G ATmega128

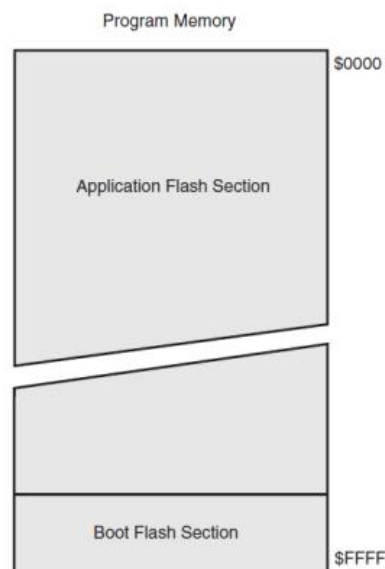
Port Pin	Fungsi Alternatif
PG4	TOSC1 (RTC Oscillator Timer/Counter0)
PG3	TOSC2 (RTC Oscillator Timer/Counter0)
PG2	ALE (Address Latch Enable to external memory)
PG1	RD (Read strobe to external memory)
PG0	WR (Write strobe to external memory)

(Sumber: Data Sheet ATmega128)

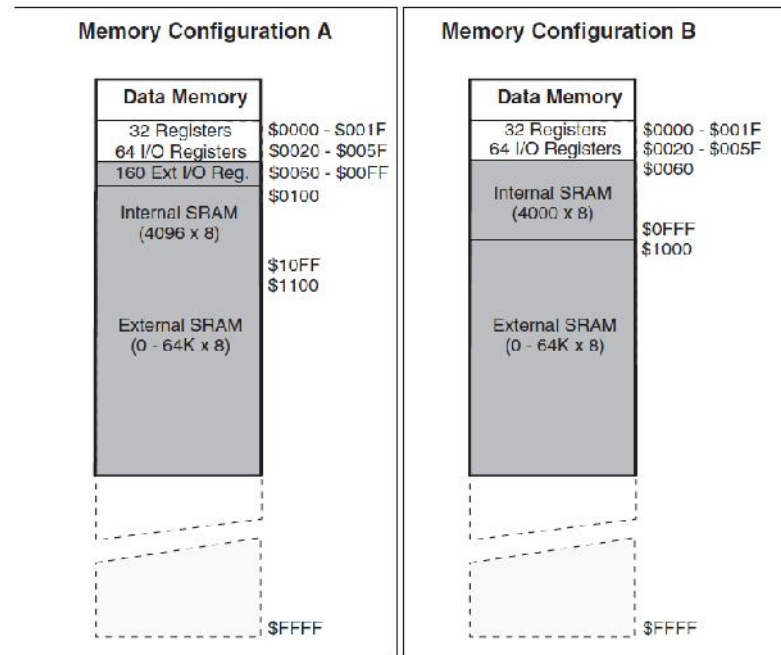
- j. RESET merupakan pin yang digunakan untuk mereset mikrokontroler.
- k. XTAL1 merupakan pin masukan untuk penguatan *inverting* pada *oscillator* dan pin masukan untuk *internal clock operating circuit*.

- l. XTAL2 merupakan keluaran *inverting Oscillator amplifier*.
 - m. AVCC merupakan pin masukkan tegangan untuk ADC.
 - n. AREF merupakan pin masukkan tegangan referensi ADC.
3. Peta Memori

Arsitektur AVR ATmega128 memiliki ruang pengalamatan memori data dan memori program yang terpisah. Selain itu, ATmega128 memiliki fitur sebuah memori EEPROM yang berfungsi untuk penyimpanan data. Penyimpanan data dalam chip mampu menyimpan data hingga 128 Kbytes pada sistem *reprogrammable flash memory*. Karena semua instruksi AVR memiliki lebar 16 bit atau 32 bit, *flash* disusun sebesar 64K x 16. Untuk keamanan perangkat lunak, ruang memori *Program Flash* dibagi menjadi dua bagian yaitu *boot* bagian program dan bagian program aplikasi yang dapat dilihat pada gambar 14 dan gambar 15. *Flash* memori memiliki daya tahan setidaknya 10.000 *write*.



Gambar 14. Memori Program AVR ATmega128
(Sumber: Datasheet ATmega128)



Gambar 15. Konfigurasi memori data AVR ATmega128
(Sumber: Datasheet ATmega128)

ATmega128 mempunyai lima mode pengalamatan untuk pengalamatan memori data yaitu, langsung, tidak langsung dengan pengungsian, langsung, tidak langsung dengan *pre-decrement*, dan tidak langsung dengan pasca kenaikan. Dalam file *register*, *register* R26 sampai R31 tidak langsung menangani pengalamatan pointer register. Memori data ATmega128 terbagi menjadi 32 register, 64 register I/O, dan 4096 *bytes* SRAM data internal yang seluruhnya dapat diakses melalui semua mode pengalamatan.

4. Status Register

Status Register adalah *register* berisi status yang dihasilkan pada setiap operasi yang dilakukan ketika suatu instruksi dieksekusi. Instruksi ini dapat digunakan untuk mengubah program yang mengalir untuk melakukan operasi bersyarat. SREG merupakan bagian inti dari inti CPU mikrokontroler. Urutan status register ATmega128 dapat dilihat pada Gambar 16.

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 16. Status Register AVR ATmega128
(Sumber: Data sheet ATmega128)

a. Bit 7-1: *Global Interrupt Enable*

Bit harus diset untuk mengaktifkan interupsi, setelah itu dapat diaktifkan interupsi mana yang akan digunakan dengan cara mengaktifkan bit kontrol register yang bersangkutan secara individu. Bit akan dikenakan operasi *clear* apabila terjadi suatu interupsi yang dipicu oleh *hardware* dan bit tidak mengijinkan terjadinya interupsi serta akan diset kembali oleh instruksi RETI untuk memungkinkan interupsi berikutnya. Selain itu bit juga dapat mengatur dan dibersihkan dalam perangkat lunak dengan instruksi SEI dan CLI.

b. Bit 6-T: *Bit Copy Storage*

Instruksi BLD (*Bit Load*) dan BST (*Bit Store*) menggunakan bit-T sebagai sumber atau tujuan dalam operasi bit. Suatu bit dari sebuah file register dapat disalin ke bit T menggunakan instruksi BST dan sebaliknya bit-T dapat disalin ke dalam setiap bit dalam file register menggunakan instruksi BLD.

c. Bit 5-H: *Half Carry Flag*

The *Half Carry Flag* menunjukkan *half carry* dalam beberapa operasi aritmatika. *Half carry* digunakan dalam BCD aritmatika.

d. Bit 4-S: *Sign Bit*

Bit-s selalu berada diantara flag-N (negatif) dan flag V (komplemen dua *overflow*).

e. Bit 3-V: *Two's Complement Overflow Flag*

Bit berguna untuk mendukung 2 komplemen operasi aritmatika.

f. Bit 2-N: *Negative Flag*

The Negative Flag N menghasilkan bilangan *negative* atau *logic operation*.

g. Bit 1-Z: *Zero Flag*

The Zero Flag Z menghasilkan nilai 0 (nol) dalam aritmatika atau *logic operation*.

h. Bit 0-C: *Carry Flag*

The Carry Flag C menghasilkan *carry* dalam aritmatika atau *logic operation*.

L. Bahasa C

Dikembangkan pertama kali oleh *Dennis Ritchie* dan *Ken Thomson* pada tahun 1972, bahasa C merupakan salah satu bahasa pemrograman yang paling populer untuk mengembangkan program-program aplikasi yang berjalan pada sistem mikroprosesor. Karena kepopulerannya dan banyak digunakan, vendor-vendor perangkat lunak kemudian mengembangkan *compiler C* sehingga menjadi beberapa varian, macam-macam varian tersebut yaitu: Turbo C, Borland C, Microsoft C, Power C, Zortech C, dan lain sebagainya. Untuk menjaga probabilitas, *compiler-compiler C* tersebut menerapkan ANSI C (*American National Standards Institute*) sebagai standar bakunya. Perbedaan antara

compiler-compiler tersebut umumnya hanya pada pengembangan fungsi-fungsi pustaka serta fasilitas IDE (*Integrated Development Environment*).

Bahasa C relatif merupakan bahasa pemrograman yang sangat fleksibel dan tidak terlalu terikat dengan berbagai aturan yang sifatnya kaku. Salah satu hal yang membatasi penggunaan bahasa C dalam sebuah aplikasi adalah semata-mata kemampuan imajinasi pemrogramnya saja. Sebagai contoh, dalam bahasa program C kita dapat saja secara bebas menjumlahkan karakter huruf (misal 'D') dengan bilangan bulat (misal '4'), dimana hal ini tidak mungkin dapat dilakukan dengan menggunakan bahasa tingkat tinggi lainnya. Karena sifatnya ini, maka bahasa C seringkali dikategorikan sebagai bahasa tingkat menengah (*mid level language*).

Dalam kaitannya dengan pemrograman mikrokontroler, bahasa C sekarang mulai menggeser bahasa yang lebih dulu digunakan untuk pemrograman mikrokontroler yaitu bahasa *assembler*. Penggunaan bahasa C akan sangat efisien terutama untuk program mikrokontroler yang berukuran relatif besar. Dibandingkan dengan bahasa *assembler*, penggunaan bahasa C dalam pemrograman memiliki beberapa kelebihan berikut. Mempercepat waktu pengembangan, bersifat modular dan terstruktur, namun kelemahannya adalah kode program hasil kompilasi akan relatif lebih besar dan sebagai konsekuensinya hal ini terkadang akan mengurangi kecepatan eksekusi.

Khusus pada mikrokontroler AVR, untuk mereduksi konsekuensi negatif diatas, Perusahaan Atmel merancang sedemikian rupa sehingga arsitektur AVR ini efisien dalam *mendekode* serta mengeksekusi instruksi-instruksi yang umum

dibangkitkan oleh *compiler* C. Dalam kenyataannya, pengembangan arsitektur AVR tidak dilakukan sendiri oleh perusahaan Atmel tetapi ada kerja sama dengan salah satu vendor pemasok *compiler* C untuk mikrokontroler tersebut, yaitu IAR C.

Beberapa *compiler* C untuk mikrokontroler AVR dapat dilihat pada Tabel 8.

. Tabel 8. Beberapa *compiler* C untuk mikrokontroler AVR

Compiler C	Platform	Keterangan
IAR C	-DOS -Windows	Komersil
CodeVisionAVR	-Windows	Komersil
ImageCraft's C	-DOS -Windows -Linux	Komersil
AVR-GCC	-DOS -Windows	General Public Licence
C-AVR	-Windows	Komersil
Small C for AVR	-DOS	Komersil
GNU C for AVR	-Linux	General Public Licence
LCC-AVR	-Linux, -Windows	Free
Dunfields AVR	-Windows	Komersil

Struktur penulisan bahasa C secara umum terdiri atas empat blok, yaitu:

1. Header.
2. Deklarasi konstanta global atau variabel.
3. Fungsi atau prosedur.
4. Program utama.

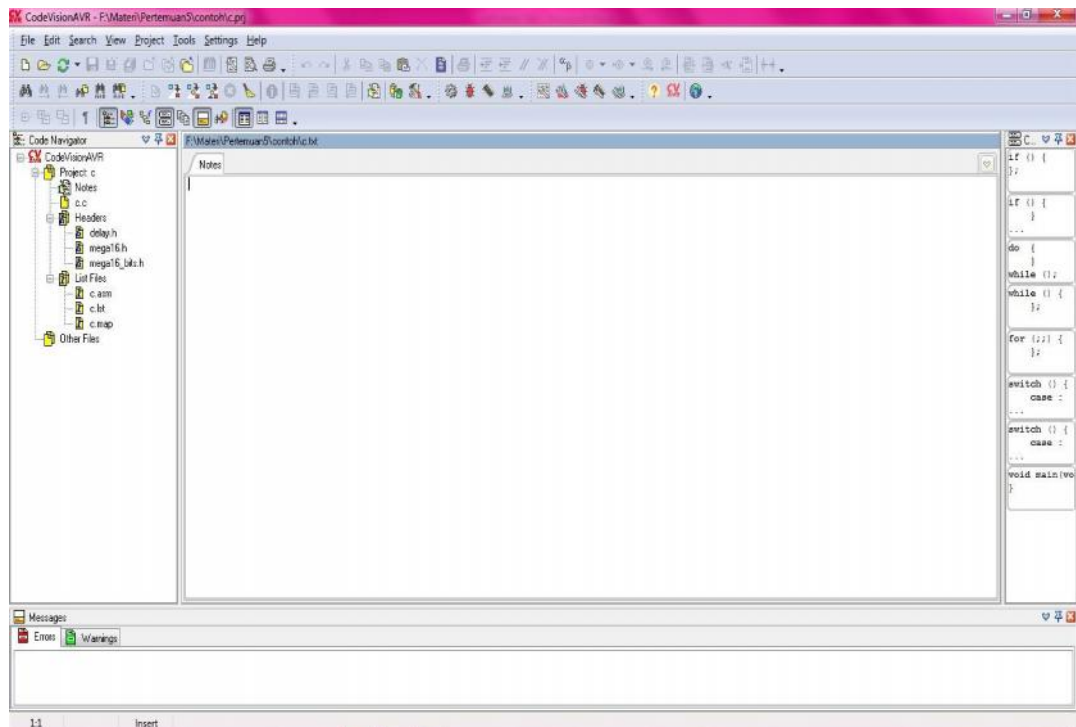
M. Perangkat Lunak CodeVision AVR (CVAVR)

CodeVision AVR pada dasarnya merupakan perangkat lunak pemrograman mikrokontroler keluarga AVR berbahasa C. terdapat tiga komponen penting yang telah diintegrasikan dalam perangkat lunak ini: *compiler* C, IDE dan Program generator.

Berdasarkan spesifikasi yang dikeluarkan oleh perusahaan pengembangannya, *compiler* C yang digunakan hampir mengimplementasikan semua komponen

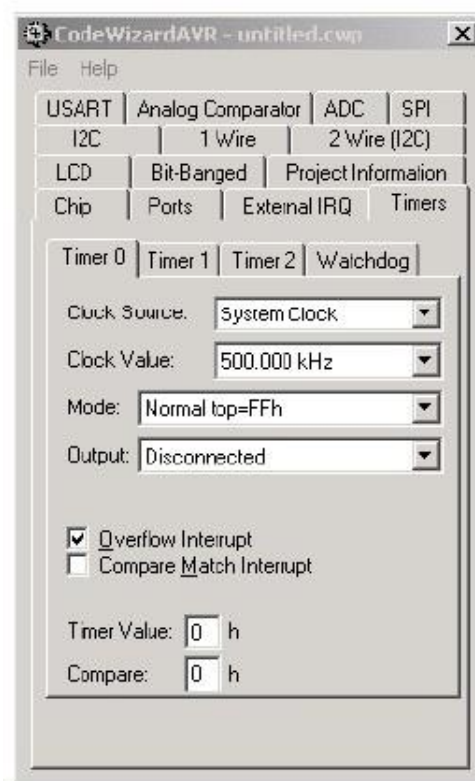
standar yang ada pada bahasa C standar ANSI (misal, struktur program, tipe data, jenis operator, dan pustaka fungsi standar-beserta penamaannya). Tetapi walau demikian, jika dibandingkan dengan bahasa C untuk aplikasi komputer, *compiler C* untuk mikrokontroler memiliki sedikit perbedaan yang disesuaikan dengan arsitektur AVR tempat C tersebut ditanamkan (*embedded*).

Khusus untuk pustaka fungsi, di samping pustaka standar (seperti fungsi-fungsi matematik, manipulasi *string*, pengaksesan memori dan sebagainya), CodeVision AVR juga menyediakan fungsi-fungsi tambahan yang sangat bermanfaat dalam pemrograman antarmuka AVR dengan perangkat luar yang umum digunakan dalam aplikasi kontrol. Beberapa fungsi pustaka yang penting diantaranya adalah fungsi-fungsi untuk pengaksesan LCD, komunikasi IC, IC RTC (*Real time Clock*), sensor suhu LM35, SPI (*Serial Peripheral Interface*) dan lain sebagainya. Untuk memudahkan pengembangan program aplikasi, CodeVision AVR juga dilengkapi IDE yang sangat *user friendly* (lihat Gambar 17). Selain menu-menu pilihan yang umum dijumpai pada setiap perangkat lunak berbasis Windows, CodeVision AVR ini telah mengintegrasikan perangkat lunak *downloader (in system programmer)* yang dapat digunakan untuk mentransfer kode mesin hasil kompilasi kedalam sistem memori mikrokontroler AVR yang sedang diprogram. Tampilan IDE perangkat lunak CodeVisionAVR dapat dilihat pada Gambar 17.



Gambar 17. IDE perangkat lunak CodeVisionAVR

CodeVisionAVR juga menyediakan sebuah *tool* yang dinamakan dengan *Code Generator* atau *CodeWizardAVR*. Secara praktis, *tool* ini sangat bermanfaat membentuk sebuah kerangka program (*template*), dan juga memberi kemudahan bagi programmer dalam peng-inisialisasian register-register yang terdapat pada mikrokontroler AVR yang sedang diprogram. Dinamakan kode generator, karena perangkat lunak CodeVision ini akan membangkitkan kode-kode program secara otomatis setelah fase inisialisasi pada jendela *CodeWizardAVR* selesai dilakukan. Secara teknis, penggunaan *tool* ini pada dasarnya hampir sama dengan *application wizard* pada bahasa-bahasa pemrograman visual untuk komputer (seperti Visual C, Borland Delphi, dan sebagainya). Kode generator yang dapat digunakan untuk menginisialisasi register-register pada mikrokontroler AVR dapat dilihat pada Gambar 18.



Gambar 18. Kode generator yang dapat digunakan untuk menginisialisasi register-register pada mikrokontroler AVR.

N. UBEC

UBEC (*Universal Battery Elimination Circuit*) adalah suatu modul rangkaian elektronik yang berfungsi sebagai regulator (penstabil tegangan). Pada umumnya rangkaian regulator menggunakan IC 7805 yang memiliki kemampuan menangani arus hingga 1A. UBEC yang mengambil daya dari *battery* atau sumber DC yang lain, dan menurunkan tegangan ke level tegangan 5V atau 6V dengan arus sesuai spesifikasi yang dimiliki UBEC. Bentuk fisik UBEC dapat dilihat pada Gambar 19.



Gambar 19. Bentuk fisik UBEC

(Sumber : <http://www.brontoseno.com/kategori-produk/sbecubec/>)

O. LCD Display 16x2 M1632

M1632 adalah modul LCD dengan tampilan 16 kolom x 2 baris dengan konsumsi daya yang rendah. Modul ini dilengkapi dengan mikrokontroler yang didesain sedemikian rupa khusus untuk mengendalikan LCD. Mikrokontroler HD44780 buatan Hitachi yang berfungsi sebagai pengendali LCD ini mempunyai CGROM (*Character Generator Read Only Memory*), CGRAM (*Character Generator Random Access Memory*) dan DDRAM. Bentuk fisik LCD dapat dilihat pada Gambar 20.



(*Display Data Random Access Memory*).

Gambar 20. LCD M1632

(Sumber : <http://www.brontoseno.com/kategori-produk/lcd-LED-display/>)

P. Motor Planetary Gear (PG) 45

Motor Planetary Gear (PG) 45 merupakan type motor DC brushless produksi Oriental Motor corp. Motor ini sudah dilengkapi gearbox dengan perbandingan yang bervariasi. Motor Planetary Gear (PG) 45 memiliki kecepatan maksimum 500 rpm. Dengan internal gearnya yang 10:1 maka kecepatan maksimum pada poros keluaran adalah 50 rpm, dengan torsi rata-rata sebesar 11,2 kg-cm. Motor ini sesuai diaplikasikan sebagai actuator robot. Motor ini membutuhkan catu 24V. Catu sebesar itu sangat sesuai dengan peraturan Kontes Robot Indonesia. Untuk mengatur kecepatan dan arah putaran motor digunakan rangkaian elektronik penggerak motor (driver motor). Bentuk rangkaian elektronik penggerak motor yang digunakan adalah H-Bridge. lihat Lampiran 6. Bentuk fisik motor *planetary gear* dapat dilihat pada Gambar 21.



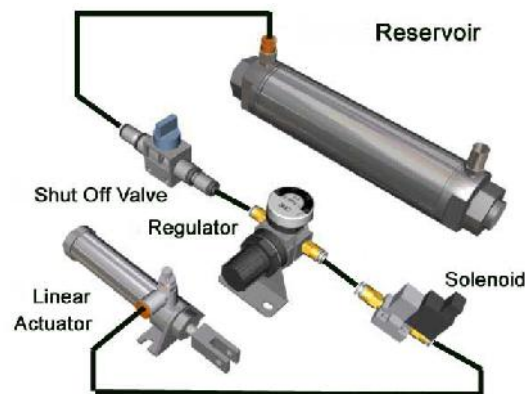
Gambar 21. Motor *planetary gear*

(Sumber: <http://www.brontoseno.com/produk/pg42mzy40geared-motor-series/>)

Q. Pneumatik

Pneumatik merupakan teori atau pengetahuan tentang udara yang bergerak, keadaan-keadaan keseimbangan udara dan syarat-syarat keseimbangan. Perkataan

pneumatik berasal bahasa Yunani “pneuma” yang berarti “napas” atau “udara”. Jadi pneumatik berarti terisi udara atau digerakkan oleh udara mampat. Pneumatik merupakan cabang teori aliran atau mekanika fluida dan tidak hanya meliputi penelitian aliran-aliran udara melalui suatu sistem saluran, yang terdiri atas pipa-pipa, selang-selang, gawai dan sebagainya, tetapi juga aksi dan penggunaan udara mampat. Pneumatik menggunakan hukum-hukum aeromekanika, yang menentukan keadaan keseimbangan gas dan uap (khususnya udara atmosfer) dengan adanya gaya-gaya luar (aerostatika) dan teori aliran (aerodinamika). Pneumatik dalam pelaksanaan teknik udara mampat dalam industri merupakan ilmu pengetahuan dari semua proses mekanik dimana udara memindahkan suatu gaya atau gerakan. Jadi pneumatik meliputi semua komponen mesin atau peralatan, dalam mana terjadi proses-proses pneumatik. Dalam bidang kejuruan teknik pneumatik dalam pengertian yang lebih sempit lagi adalah teknik udara mampat (udara bertekanan). Bentuk fisik sederhana pneumatik dapat dilihat pada Gambar 22.



Gambar 22. Pneumatik

(Sumber: <http://www.graphicwebparts.com/part/166876908/167-040-200-0/rexroth-mecman-pneumatik/pneumatics/pneumatic-cylinder>)

Komponen-komponen Pneumatik Komponen pneumatik beroperasi pada tekanan 8 s.d. 10 bar, tetapi dalam praktik dianjurkan beroperasi pada tekanan 5 s.d. 6 bar untuk penggunaan yang ekonomis. Beberapa bidang aplikasi di industri yang menggunakan media pneumatik dalam hal penanganan material adalah sebagai berikut.

1. Pencekaman benda kerja
2. Penggeseran benda kerja
3. Pengaturan posisi benda kerja
4. Pengaturan arah benda kerja

Penerapan pneumatik secara umum:

1. Pengemasan (packaging)
2. Pemakanan (feeding)
3. Pengukuran (metering)
4. Pengaturan buka dan tutup (door or chute control)
5. Pemindahan material (transfer of materials)
6. Pemutaran dan pembalikan benda kerja (turning and inverting of parts)
7. Pemilahan bahan (sorting of parts)
8. Penyusunan benda kerja (stacking of components)
9. Pencetakan benda kerja (stamping and embossing of components)

Susunan sistem pneumatik adalah sebagai berikut.

1. Catu daya (energi supply)
2. Elemen masukan (sensors)
3. Elemen pengolah (processors)

4. Elemen kerja (actuators)

Dalam penggunaan pneumatik diperlukan rangkaian elektronik penggerak pneumatik yang merupakan rangkaian khusus dengan fungsi mengatur arah tekanan angin dengan menggerakkan katup *solenoid*. Adanya rangkaian ini karena pada ATmega tidak bisa memberikan arus lebih dari 250 mA dan tegangan lebih dari 6 volt ke *solenoid*, sehingga diperlukan rangkaian elektronik yang dapat memberikan arus dan tegangan yang lebih besar dari keluaran Atmega untuk menggerakkan katup *solenoid*. Apabila *solenoid* langsung dihubungkan ke ATmega, maka hal ini dapat menyebabkan kerusakan pada Atmega tersebut. Lihat lampiran 7. Rangkaian elektronik penggerak *solenoid*.

BAB III

PERANCANGAN DAN PEMBUATAN ALAT

Perancangan robot bulutangkis ini beracuan pada peraturan Kontes Robot Abu Indonesia 2015 yang meliputi dimensi robot, berat robot, catu daya dan peraturan pertandingan. Berdasarkan peraturan tersebut, metode rancang bangun merupakan metode yang tepat. Secara berurutan metode tersebut adalah identifikasi kebutuhan yang diperlukan. Kemudian kebutuhan tersebut dianalisis untuk mendapatkan komponen secara spesifik. Selanjutnya dilakukan perancangan perangkat keras dan perangkat lunak, pembuatan serta pengujian.

A. Identifikasi Kebutuhan

Identifikasi kebutuhan sebagai berikut.

1. Tumpuan utama pada pembuatan robot bulutangkis adalah roda robot, sehingga dibutuhkan mekanik roda dan motor yang sesuai.
2. Penggerak roda dan raket berupa motor dan pneumatik perlu konsumsi daya yang cukup agar torsi dan arus terpenuhi, sehingga dibutuhkan driver (rangkaian) untuk penggerak motor pada robot bulutangkis.
3. Sebuah robot agar dapat bergerak dibutuhkan pengontrol gerakan yang sesuai.
4. Sebuah robot perlu pengaturan yang tepat, sehingga diperlukan pengaturan pergerakan robot dengan tampilan pengaturan pada LCD.
5. Sebuah robot perlu seorang *driver/operator* yang tepat, sehingga diperlukan latihan intensif seorang driver guna mengontrol robot dengan baik.

B. Analisis Kebutuhan

Berdasarkan dari beberapa identifikasi kebutuhan di atas maka diperoleh beberapa analisis kebutuhan terhadap pengembangan alat yang akan dibuat sebagai berikut.

1. Rangkaian catu daya sebagai penyedia tegangan diambil dari 2 buah *lithium polymer battery* 11,1 Volt 1 ampere sebagai penyedia tegangan sistem minimum.
2. Rangkaian catu daya sebagai penyedia tegangan diambil dari 1 buah *lithium polymer battery* 22,2 Volt 2,2 ampere sebagai penyedia tegangan motor dan Pneumatik, dalam hal ini menggunakan modul UBEC.
3. Rangkaian sistem minimum menggunakan ATmega128, digunakan untuk mengontrol seluruh motor dan Pneumatik yang digunakan sebagai penggerak roda dan penggerak katup silinder, karena dinilai sangat efisien dan memiliki kecepatan yang sesuai sebagai kontrol motor dan pneumatik, dengan kata lain rangkaian ini disebut motor kontroller.
4. Rangkaian sistem minimum menggunakan ATmega16, digunakan untuk mengontrol seluruh kerja dari robot bulutangkis karena dinilai sangat praktis, kemudian dihubungkan ke Aktuator.
5. Motor digunakan untuk mengerakkan roda-roda robot karena mudah dalam pengendalian.
6. Motor dan Pneumatik digunakan untuk menggerakkan raket saat servis dan mengembalikan servis.

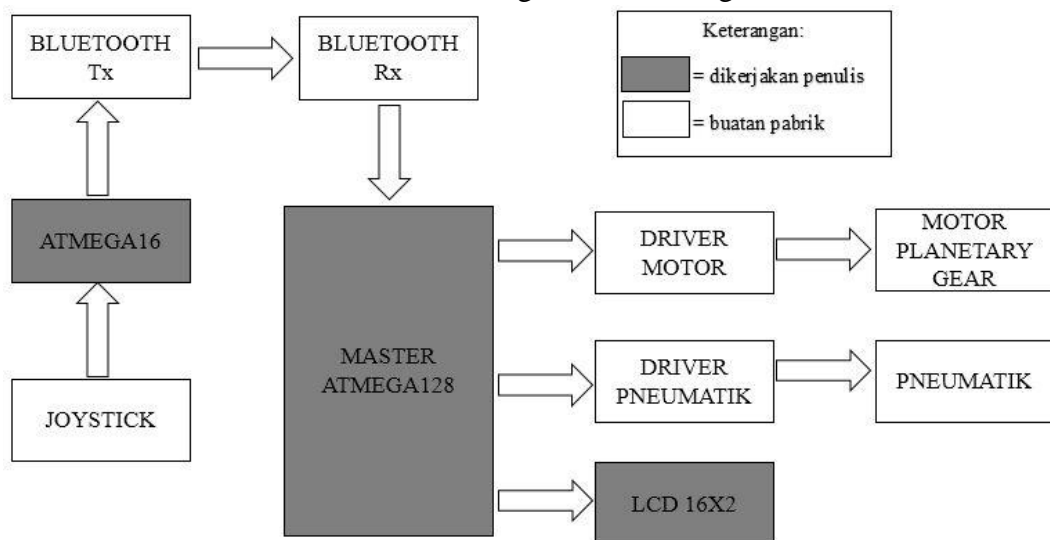
7. Penambahan LCD M1632 sebagai penampil keadaan sistem dan *joystick* sebagai tombol masukan pengaturan sistem.
8. Driver sebagai orang yang mengendalikan robot dan mengerti kekurangan serta kelebihan robot.

Blok Diagram Rangkaian

Blok diagram utama rangkaian dan spesifikasi blok diagram utama rangkaian dapat dilihat pada Gambar 23 dan 24.



Gambar 23. Blok diagram utama rangkaian







Gambar 24. Spesifikasi blok diagram utama rangkaian

Keterangan blok diagram diatas:







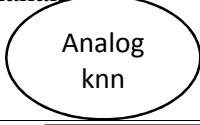






1. Input joystick digunakan untuk memilih menu, menentukan besar nilai variabel sebuah konstanta dan masuk ke program utama.

2. Bluetooth digunakan sebagai komunikator yaitu *transmitter* (Tx) pada joystick dan *receiver* (Rx) pada robot bulutangkis.
3. Pneumatik yang difungsikan sebagai penggerak raket, Untuk meng-ON-kan raket dengan menggunakan solenoid dengan *trigger* langsung dari mikrokontroler.
4. Motor sebagai keluaran untuk mekanik roda. Motor untuk mekanik roda berjumlah 4 buah. Motor ini terhubung dengan IC ATmega128. ATmega128 disini khusus sebagai *driver* motor mekanik roda.
5. LCD sebagai monitor kegiatan sisem seperti menu, kondisi sensor, dan kondisi khusus. LCD ini terhubung dengan PORT C pada mikrokontroler ATmega128.
6. Programming *device* digunakan untuk memprogram mikrokontroler atau memasukkan *memory flash* atau *eprom* berupa file .hex dan file.eep dari komputer ke mikrokontroler.

Tabel 9. Konfigurasi tombol pada joystick *wireless*

No	Tombol	Fungsi
1	X = 	Untuk menggerakkan motor yang terhubung dengan raket saat servis
2	Segitiga = 	Untuk menggerakkan motor servis
3	Persegi = 	Untuk membalikkan arah putaran motor setelah servis
4	Lingkaran = 	Untuk menggerakkan pneumatik yang terhubung dengan raket

Lanjutan Tabel 9

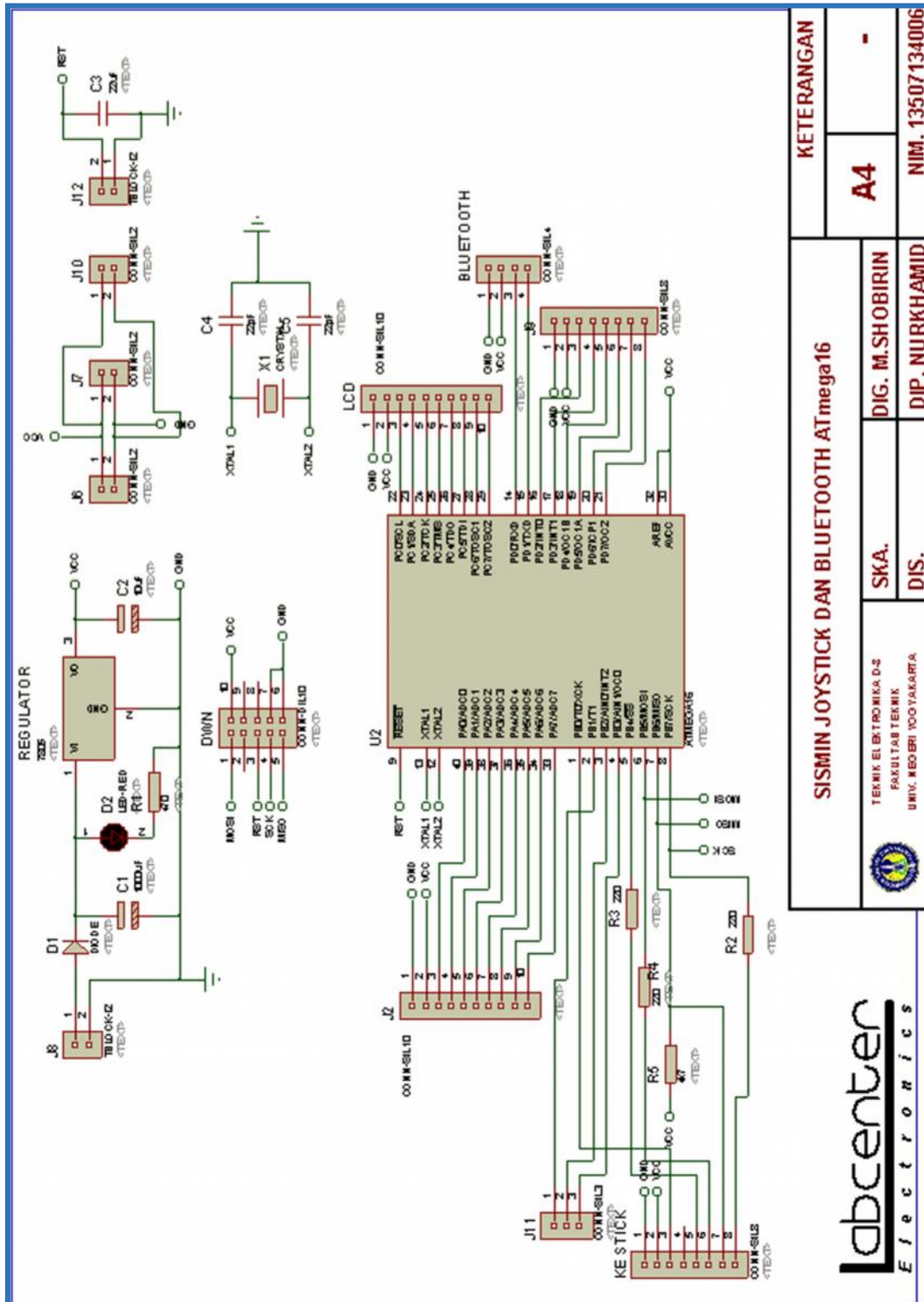
No	Tombol	Fungsi
5	Start = 	Untuk masuk ke pengaturan program
6	Select = 	Untuk menampilkan menu program
7	Analog on/off = 	Untuk menghidupkan fungsi semua tombol
8	R1= 	Untuk mempercepat putaran motor base
9	R2= 	Untuk memperlambat putaran motor base
10	Analog kiri= 	Untuk menggerakkan motor base sesuai arah yang diinginkan
11	Analog kanan= 	Untuk memutar bagian base robot dengan menggerakkan motor base
12	L1= 	Untuk membuka griper kok saat servis
13	L2= 	Untuk menutup griper kok saat servis
14	Arah kanan= 	Untuk mengarahkan memilih menu ke kanan
15	Arah kiri= 	Untuk mengarahkan memilih menu ke kiri
16	Arah bawah= 	Untuk mengarahkan memilih menu ke atas
17	Arah atas= 	Untuk mengarahkan memilih menu ke bawah

Tabel 9 merupakan daftar tombol konfigurasi joystick yang digunakan. Sebagai contoh gerakan servis robot maka driver harus menekan tombol kombinasi **L1 Tahan dan kemudian X** , maka gerakan yang terjadi adalah *griper* yang memegang kok akan membuka kemudian raket yang terhubung pada motor akan memukul kok ke arah tujuan dan berputar kembali ke arah sebaliknya. Untuk menggerakkan maju, mundur, dan samping menggunakan **analog kiri**.

Gambar 25 merupakan gambar Rangkaian sistem minimum ATmega128. Pada sistem minimum ATmega128 ini menggunakan komunikasi secara paralel. PORTA difungsikan untuk menggerakkan motor 2 dan 4 dengan pwm. PORTE difungsikan untuk menggerakkan motor 1 dan 3 dengan pwm. Motor 1, 2, 3, dan 4 digunakan sebagai penggerak base robot. PIN D2 dan D3 difungsikan untuk komunikasi serial menggunakan bluetooth. PIN D4 dan D5 difungsikan untuk menggerakkan pneumatik besar dan kecil pneumatik besar difungsikan sebagai penggerak raket dan pneumatik kecil. Sebagai penjepit kok saat akan servis. PORTF difungsikan untuk menggerakkan motor 5 dengan pwm. Motor 5 difungsikan untuk menggerakkan raket saat servis dan membalikkan servis lawan. PORTC difungsikan untuk menampilkan karakter menu saat pengaturan dan menjalankan program.

MOSI, MISO, SCK, RST, GND, dan VCC dibuat dalam pin IDC. Untuk memudahkan fungsi download program ke ATmega128. Rangkaian regulator terpisah dari sismin ATmega128 berfungsi sebagai catu daya. Rangkaian regulator terdiri dari ic 7805, kapasitor, resistor, LED, dioda, dan konektor 2 pin. Ic regulator 7805 berfungsi sebagai penstabil tegangan dan mengeluarkan tegangan 4,75-5V. Inputan ic 7805 berkisar 7-25V. Dioda difungsikan sebagai pengaman tegangan umpan balik. Pemasangan kapasitor sebelum dan sesudah ic 7805 berfungsi sebagai filter/penyaring tegangan. Resistor difungsikan menghambat/mengurangi arus yang masuk pada LED agar tidak berlebihan. LED difungsikan sebagai indikator adanya tegangan masuk. Konektor 2 pin sebagai masukan vcc dan gnd.

2. Rangkaian Sistem minimum ATmega16



Gambar 26. Rangkaian Sistem Minimum ATmega16

Gambar 26 merupakan gambar Rangkaian sistem minimum ATmega16. PORTB terpasang pin untuk joystick difungsikan untuk memberikan masukan perintah pada program yang telah terdownload pada ATmega16. PIN D0 dan D1 difungsikan untuk komunikasi serial menggunakan bluetooth. PORTC difungsikan untuk menampilkan karakter menu saat pengaturan dan menjalankan program.

MOSI, MISO, SCK, RST, GND, dan VCC dibuat dalam pin IDC. Untuk memudahkan fungsi download program ke ATmega128. Rangkaian regulator terpisah dari sismin ATmega128 berfungsi sebagai catu daya. Rangkaian regulator terdiri dari ic 7805, kapasitor, resistor, LED, dioda, dan konektor 2 pin. Ic regulator 7805 berfungsi sebagai penstabil tegangan dan mengeluarkan tegangan 4,75-5V. Inputan ic 7805 berkisar 7-25V. Dioda difungsikan sebagai pengaman tegangan umpan balik. Pemasangan kapasitor sebelum dan sesudah ic 7805 berfungsi sebagai filter/penyaring tegangan. Resistor difungsikan menghambat/mengurangi arus yang masuk pada LED agar tidak berlebihan. LED difungsikan sebagai indikator adanya tegangan masuk. Konektor 2 pin sebagai masukan vcc dan gnd.

D. Langkah Pembuatan Alat

Langkah pembuatan alat pada proyek akhir ini terdiri dari pembuatan mekanik casis, pelarutan PCB, pemasangan komponen pada PCB dan peletakan aktuator.

1. Pembuatan Mekanik Robot

a. Perencanaan Ukuran Mekanik Robot

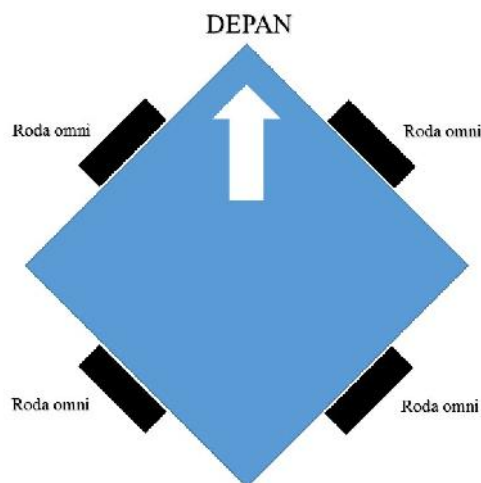
Panjang : 150 cm

Lebar : 150 cm

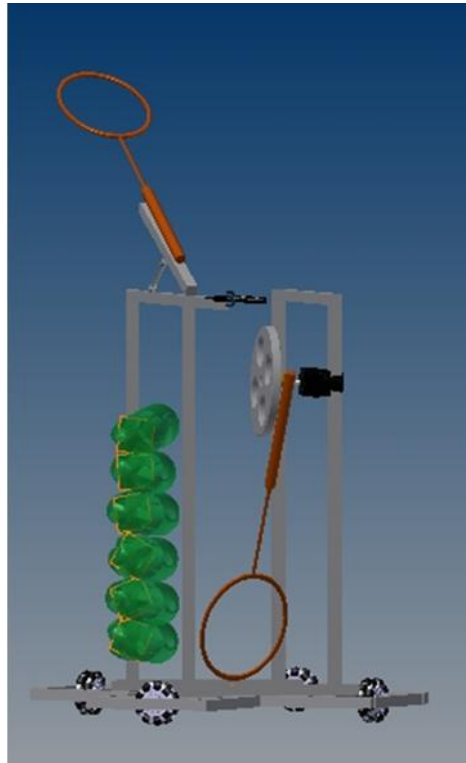
Tinggi : 150 cm

b. Perencanaan Mekanik Robot

Proses pembuatan mekanik robot dilakukan di bengkel dengan peralatan seadanya. Bagian mekanik roda dan bagian rangka digunakan untuk memasang modul rangkaian. Proses pembuatan mekanik dilakukan dari desain mekanik, pemotongan aluminium sebagai bahan dasar mekanik, plat dan penyesuaian perakitan. Dari percobaan-percobaan yang dilakukan didapatkan hasil bentuk roda pada base dan rangka terlihat pada Gambar 27.



Gambar 27. Rancangan peletakan roda pada base robot



Gambar 28. Rancangan mekanik robot

Gambar 28 Merupakan gambar rancangan mekanik robot dengan dua buah raket.

2. Pembuatan PCB

a. Pembuatan *layout* PCB

Langkah awal pembuatan PCB adalah menggambar *layout* rangkaian dengan perangkat lunak. Perangkat lunak yang digunakan yaitu Proteus. Hasil penggambaran *layout* PCB dapat dilihat pada Lampiran 1.

b. Penyablonan PCB

Setelah *layout* selesai dibuat maka langkah selanjutnya yaitu penyablonan, atau sering disebut pembuatan PCB. Proses penyablonan dilakukan dengan cara sebagai berikut.

- 1) Mencetak *layout* PCB yang telah dibuat pada kertas glossi.

- 2) Desain *layout* yang sudah dicetak pada kertas glossi dipindahkan atau disablonkan ke PCB dengan cara disetrika dengan panas sedang sampai dirasa gambar sudah terpindah semua.
- 3) Setelah gambar *layout* menempel semua pada PCB maka tunggu sebentar sampai dingin, kemudian bersihkan kertas yang menempel pada PCB dengan menggunakan air bersih.

c. Pelarutan dan Pengeboran PCB

Proses melarutkan PCB atau sering disebut juga pembuatan jalur pada PCB. Langkah ini merupakan lanjutan dari proses sebelumnya, pada proses ini PCB yang sudah di sablon kemudian dilarutkan menggunakan cairan kimia HCl (*Asam Chloride*) sampai jalur rangkaian terbentuk semua. Kemudian setelah jalur PCB terbuat, mengambil PCB dari cairan HCl dan membersihkan dengan air sampai sisa tinta sablon bersih. Langkah selanjutnya yaitu pengeboran PCB, PCB di bor sesuai dengan titik titik yang telah ditentukan.

d. Pemasangan Komponen

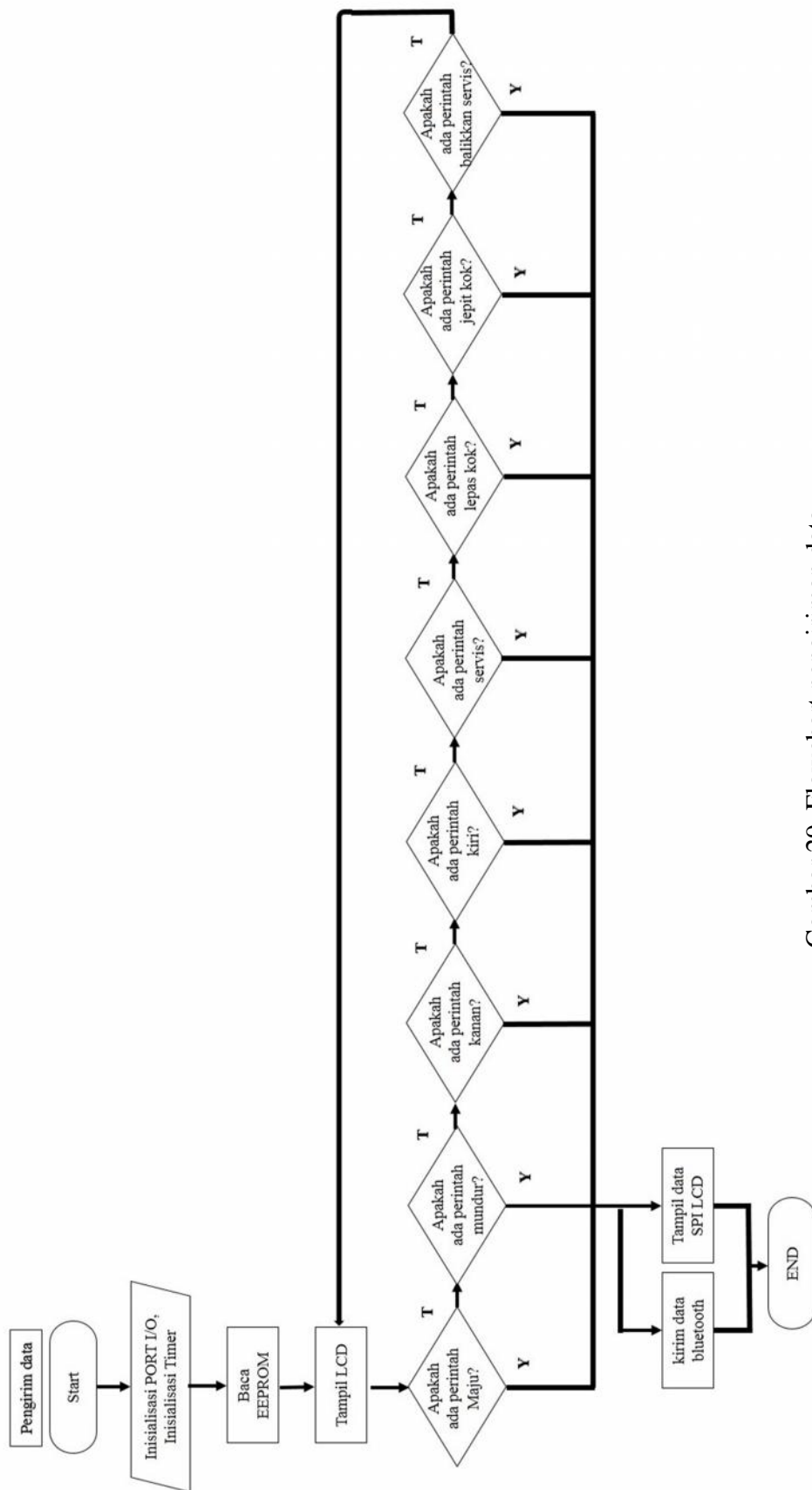
Langkah selanjutnya yaitu pemasangan seluruh komponen pada PCB, dengan urutan sebagai berikut.

- 1) Menyiapkan seluruh komponen yang dibutuhkan.
- 2) Memasang komponen dari komponen yang memiliki ukuran yang paling kecil. Terlebih pemasangan jumper didahulukan.
- 3) Menyolder kaki komponen sampai semua komponen terpasang, dan kematangan timah selalu dijaga.

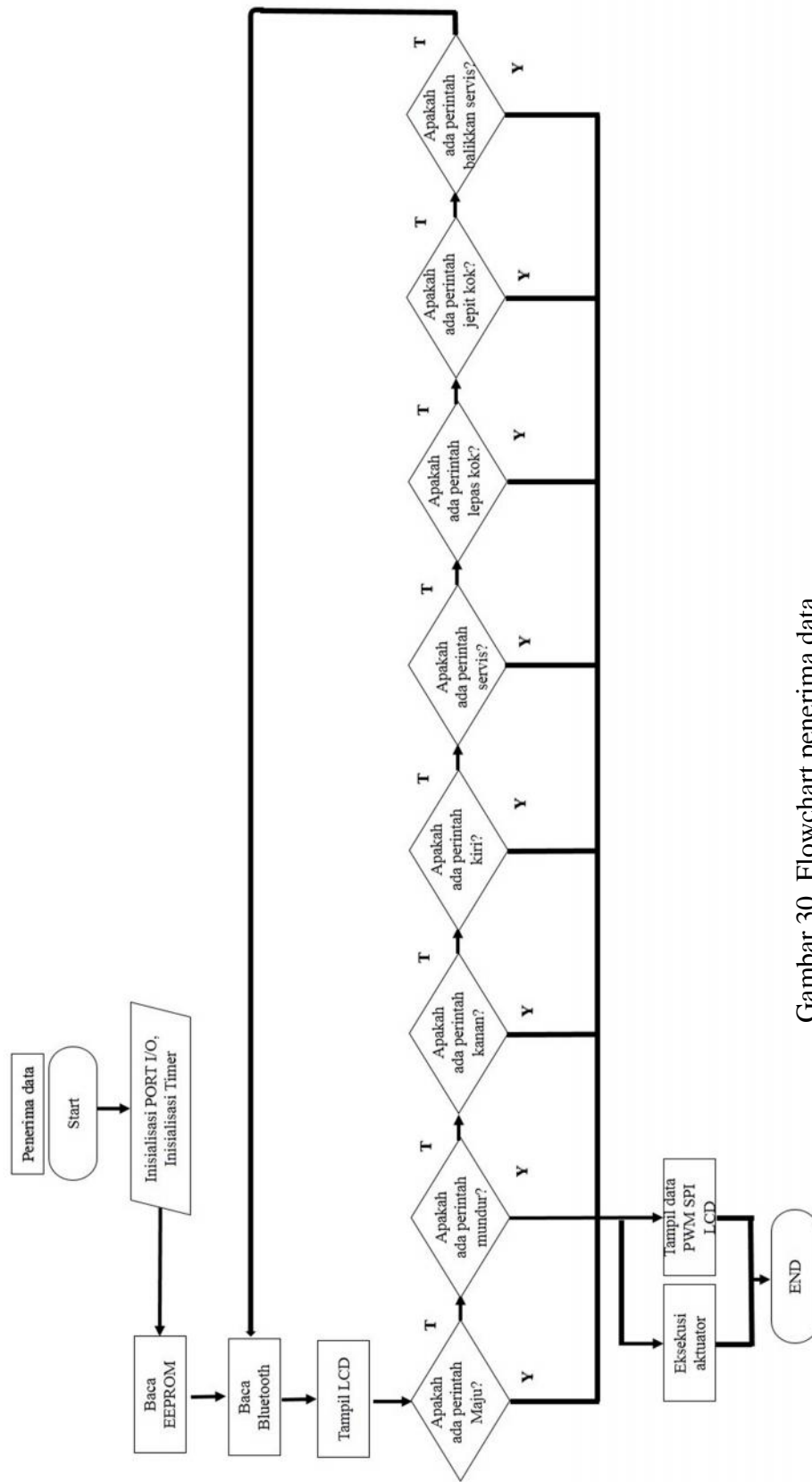
- 4) Menguji rangkaian apakah sudah dapat bekerja dengan baik atau belum.

E. Diagram Alir/*Flowchart* Program

Alur pemrograman robot bulutangkis ini dimulai dari start yang berarti awal dimulainya program. Setelah program berjalan mikrokontroler akan melakukan proses inisialisasi pada fasilitas-fasilitas yang terdapat pada mikrokontroler ATmega16 dan ATmega128 baik yang digunakan fiturnya ataupun yang tidak digunakan. Alur program secara lengkap dapat dilihat pada Gambar 29 untuk pengirim data dan Gambar 30 untuk penerima data.



Gambar 29. Flowchart pengiriman data



Gambar 30. Flowchart penerima data

F. Perancangan Program

Program pada robot bulutangkis ini dibuat menggunakan bahasa tingkat tinggi yaitu bahasa C dan menggunakan software pembantu yaitu Code Vision AVR. Dalam pembuatan program ini digunakan beberapa fasilitas yang terdapat dalam mikrokontroler ATmega16 maupun ATmega128 diantaranya yaitu *Timer*, LCD, dan RX/TX. Susunan program pada IC ATmega16 yaitu berisi dari program pengaturan menu utama, pengontrolan komunikasi paralel, dan perhitungan *timer*, sedangkan pada IC ATmega128 yaitu berisi dari program pemutaran motor, pneumatik, komunikasi paralel, dan perhitungan *timer*. Program alat ini mengacu pada beberapa sumber yaitu buku Pemrograman Mikrokontroler ATmega16 menggunakan CodeVision AVR (Heri Andrianto:2008).

G. Perencanaan Pengujian dan Pengambilan Data

Pengujian sistem dan pengambilan data dimaksudkan untuk mengetahui kinerja robot. Aktuator yang digunakan perlu diuji kerja sebelum pengambilan data keseluruhan.

1. Pengujian catu daya

Pengujian catu daya dilakukan untuk memastikan tegangan keluaran dari catu daya tidak melebihi tegangan yang diinginkan. Pada pengujian rangkaian catu daya kali ini dilakukan dengan cara mengukur tegangan output yang dihasilkan oleh catu daya. Pengukuran dilakukan sebanyak lima kali untuk memastikan apakah rangkaian catu daya asih bekerja secara optimal atau tidak. Hasil pengukuran tegangan output catu daya ditunjukkan pada Tabel 10 dan 11.

Tabel 10. Pengukuran tegangan regulator 7805 pada modul mikrokontroler ATmega16

No	Pengukuran	V Out (V)
1	I	
2	II	
3	III	
4	IV	
5	V	




Tabel 11. Pengukuran tegangan regulator 7805 pada modul mikrokontroler ATmega128

No	Pengukuran	V Out (V)
1	I	
2	II	
3	III	
4	IV	
5	V	

2. Pengujian pneumatik

Pengujian pneumatik dilakukan untuk memastikan kondisi silinder-silinder kerja ganda yang dipakai ketika tekanan udara masuk. Pada pengujian pneumatik ini dilakukan dengan cara menekan tombol pada joystick. Pengujian dilakukan sebanyak lima kali untuk memastikan sistem masih bekerja secara optimal atau tidak. Hasil pengujian pneumatik ditunjukkan pada Tabel 12.

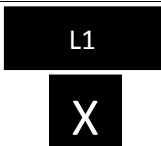
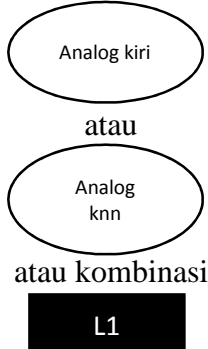
Tabel 12. Pengujian pneumatik

No.	Tombol konfigurasi	Gerakan	Percobaan ke-	Berhasil / Tidak
1.	 atau 	Membuka dan menutup penjepit kok	I	
			II	
			III	
			IV	
			V	
2.		Menggerakkan raket	I	
			II	
			III	
			IV	
			V	

3. Pengujian motor

Pengujian motor dilakukan untuk memastikan kondisi motor yang dipakai ketika berputar. Pada pengujian motor ini dilakukan dengan cara menekan tombol pada joystick. Pengujian dilakukan sebanyak lima kali untuk memastikan sistem masih bekerja secara optimal atau tidak. Hasil pengujian motor ditunjukkan pada Tabel 13.

Tabel 13. Pengujian motor

No.	Tombol konfigurasi	Gerakan	Percobaan ke-	Berhasil / Tidak
1.		Menggerakkan raket untuk servis	I	
			II	
			III	
			IV	
			V	
2.		Menggerakkan base robot	I	
			II	
			III	
			IV	
			V	

4. Pengujian komunikasi *wireless* robot

Pengujian komunikasi *wireless* robot dilakukan untuk memastikan data komunikasi yang diterima atau dikirim berpengaruh terhadap sistem. Pada pengujian komunikasi *wireless* robot ini dilakukan dengan cara menekan tombol pada joystick, kemudian mengukur jarak terdekat hingga terjauh komunikasi data yang dapat diterima. Pengujian dilakukan sebanyak lima kali untuk memastikan sistem masih bekerja secara optimal atau tidak. Hasil pengujian komunikasi *wireless* robot ditunjukkan pada Tabel 14.

Tabel 14. Pengujian Komunikasi *Wireless* Robot 1

Jarak(m)	Percobaan ke-			Respon			Komunikasi		
				I	II	III	I	II	III
1	I	II	III						
2	I	II	III						
3	I	II	III						
4	I	II	III						
5	I	II	III						
6	I	II	III						
7	I	II	III						
8	I	II	III						
9	I	II	III						
10	I	II	III						
11	I	II	III						
12	I	II	III						
13	I	II	III						
14	I	II	III						
15	I	II	III						
16	I	II	III						

BAB IV

PENGUJIAN DAN PEMBAHASAN

A. Pengujian

1. Tujuan Pengujian

Pengujian terhadap suatu alat dilakukan untuk mengetahui unjuk kerja suatu sistem secara keseluruhan. Hasil pengamatan dan pengambilan data tersebut diharapkan mampu diperoleh data yang valid.

2. Tempat Pengujian

Tempat pengujian Proyek Akhir yang berjudul “Pembuatan Sistem Kendali Robot Bulutangkis *Wireless*” ini dilakukan di dalam Aula FT UNY. Lantai Aula FT UNY adalah lantai keramik dengan ukuran masing-masing keramik adalah 40 x 40 cm. Adanya ukuran tersebut memudahkan dalam menentukan titik servis.

3. Hasil Pengujian

Berikut hasil pengujian yang telah dilakukan:

a. Pengujian catu daya

Pengujian catu daya dilakukan untuk memastikan tegangan keluaran dari catu daya tidak melebihi tegangan yang diinginkan. Pada pengujian rangkaian catu daya kali ini dilakukan dengan cara mengukur tegangan output yang dihasilkan oleh catu daya. Pengukuran dilakukan sebanyak lima kali untuk memastikan apakah rangkaian catu daya masih bekerja secara optimal atau tidak. Hasil pengukuran tegangan output catu daya ditunjukkan pada tabel 15 dan 16.

Tabel 15. Hasil Pengujian Pengukuran tegangan regulator 7805 pada modul mikrokontroler ATmega16

No	Pengukuran	V Out (V)
1	I	5,01
2	II	5,0
3	III	5,0
4	IV	5,0
5	V	4,9




Tabel 16. Hasil Pengujian Pengukuran tegangan regulator 7805 pada modul mikrokontroler ATmega128

No	Pengukuran	V Out (V)
1	I	5,03
2	II	5,03
3	III	5,0
4	IV	5,0
5	V	5,0

b. Pengujian pneumatik

Pengujian pneumatik dilakukan untuk memastikan kondisi silinder-silinder kerja ganda yang dipakai ketika tekanan udara masuk. Pada pengujian pneumatik ini dilakukan dengan cara menekan tombol pada joystick. Pengujian dilakukan sebanyak lima kali untuk memastikan sistem masih bekerja secara optimal atau tidak. Hasil pengujian pneumatik ditunjukkan pada Tabel 17.

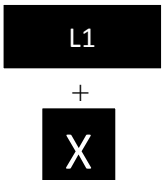

Tabel 17. Hasil pengujian pneumatik

No.	Tombol konfigurasi	Gerakan	Percobaan ke-	Berhasil / Tidak
1.	 atau 	Membuka dan menutup penjepit kok	I	Berhasil
			II	Berhasil
			III	Berhasil
			IV	Berhasil
			V	Berhasil
2.		Menggerakkan raket	I	Berhasil
			II	Berhasil
			III	Berhasil
			IV	Berhasil
			V	Berhasil

c. Pengujian motor

Pengujian motor dilakukan untuk memastikan kondisi motor yang dipakai ketika berputar. Pada pengujian motor ini dilakukan dengan cara menekan tombol pada joystick. Pengujian dilakukan sebanyak lima kali untuk memastikan sistem masih bekerja secara optimal atau tidak. Hasil pengujian motor ditunjukkan pada Tabel 18.

Tabel 18. Hasil pengujian motor

No.	Tombol konfigurasi	Gerakan	Percobaan ke-	Berhasil / Tidak
1.		Menggerakkan raket untuk servis	I	Berhasil
			II	Berhasil
			III	Berhasil
			IV	Berhasil
			V	Berhasil
2.	Analog kiri atau Analog knn atau kombinasi 	Menggerakkan base robot	I	Berhasil
			II	Berhasil
			III	Berhasil
			IV	Berhasil
			V	Berhasil

d. Pengujian komunikasi *wireless* robot

Pengujian komunikasi *wireless* robot dilakukan untuk memastikan data komunikasi yang diterima atau dikirim berpengaruh terhadap sistem. Pada pengujian komunikasi *wireless* robot ini dilakukan dengan cara menekan tombol pada joystick, kemudian mengukur jarak terdekat hingga terjauh komunikasi data yang dapat diterima. Pengujian dilakukan sebanyak lima kali

untuk memastikan sistem masih bekerja secara optimal atau tidak. Hasil pengujian komunikasi *wireless* robot ditunjukkan pada Tabel 19.

Tabel 19. Hasil pengujian komunikasi *wireless* robot

Jarak(m)	Percobaan ke-			Respon			Komunikasi		
				I	II	III	I	II	III
1	I	II	III	Baik	Baik	Baik	Cepat	Cepat	Cepat
2	I	II	III	Baik	Baik	Baik	Cepat	Cepat	Cepat
3	I	II	III	Baik	Baik	Baik	Cepat	Cepat	Cepat
4	I	II	III	Baik	Baik	Baik	Cepat	Cepat	Cepat
5	I	II	III	Baik	Baik	Baik	Cepat	Cepat	Cepat
6	I	II	III	Baik	Baik	Baik	Cepat	Cepat	Cepat
7	I	II	III	Baik	Baik	Baik	Cepat	Cepat	Cepat
8	I	II	III	Baik	Baik	Baik	Cepat	Cepat	Cepat
9	I	II	III	Baik	Baik	Baik	Cepat	Cepat	Cepat
10	I	II	III	Baik	Baik	Baik	Cepat	Cepat	Cepat
11	I	II	III	Baik	Baik	Baik	Cepat	Cepat	Cepat
12	I	II	III	Baik	Baik	Baik	Cepat	Cepat	Cepat
13	I	II	III	Baik	Baik	Baik	Cepat	Cepat	Cepat
14	I	II	III	Jelek	Jelek	Jelek	Nyambung – putus	Nyambung – putus	Nyambung – putus
15	I	II	III	Jelek	Jelek	Jelek	Nyambung – putus	Nyambung – putus	Nyambung – putus
16	I	II	III	Jelek	Jelek	Jelek	Putus	Putus	Putus

B. Pembahasan

Berdasarkan hasil pengukuran dari beberapa rangkaian dan komponen pada alat ini, dapat disimpulkan bahwa rangkaian ini dapat bekerja dengan baik dan sesuai dengan fungsinya. Pada pengukuran beberapa rangkaian sistem terdapat perbedaan sedikit dari hasil pengukuran dengan apa yang diperoleh dari teori atau dari datasheet komponen. Perbedaan hasil pengukuran terjadi karena ada beberapa faktor seperti toleransi nilai komponen dari pabrik, nilai komponen yang tidak sesuai dengan labelnya, kesalahan pengukuran dan kondisi alat ukur yang tidak bagus..

A. Analisis pengujian catu daya

Tegangan kerja sebuah mikrokontroler ATmega16 dan ATmega128 adalah 5V, maka diperlukan pencatu daya modul. Berdasarkan pada 5 kali pengukuran menggunakan multimeter digital, diperoleh tegangan 5V untuk pencatu modul mikrokontroler ATmega16 dan ATmega128. IC 7805 sebagai regulator tegangan masukan 12V memberikan keluaran +5V. Jadi dapat diketahui persentase errornya adalah sebagai berikut.

Pk = Persentase Kesalahan pengukuran

V1 = Rata-rata Tegangan Catu Daya 1

V2 = Rata-rata Tegangan Catu Daya 2

$$V1 = (5,01+5,0+5,0+5,0+4,9) / 5 = 5,0 \text{ V}$$

$$V2 = (5,03+5,03+5,0+5,0+5,0) / 5 = 5,012 \text{ V}$$

Pk1 =

$$\left(\frac{5 \text{ v} - 5 \text{ v}}{5 \text{ v}} \right) \times 100\% = 0\%$$

Pk2 =

$$\left(\frac{5,012 \text{ v} - 5 \text{ v}}{5 \text{ v}} \right) \times 100\% = 0,24\%$$

Hal-hal yang dapat mempengaruhi penurunan nilai tegangan ini di antaranya sebagai berikut.

- a. Terjadinya efek pembebanan pada rangkaian.
- b. Adanya toleransi komponen yang digunakan.
- c. Terjadi kesalahan pada saat pengukuran.
- d. Kondisi dari alat ukur yang kurang baik.

B. Analisis pengujian pneumatik

Berdasarkan analisis kebutuhan, pneumatik dibutuhkan untuk menggerakkan raket dan penjepit kok. Hasil dari 5 kali percobaan menunjukkan bahwa tegangan keluaran dari modul bernilai 5V dan mampu untuk memicu rangkaian penggerak *solenoid valve pneumatic* (katup yang digerakkan oleh cenergi listrik) sehingga tekanan udara dapat masuk dan menggerakkan piston pada silinder (kerja ganda) pneumatik keluar dan masuk dengan respon baik.

$$\frac{5}{5} \times 100\% = 100\%$$

5 kali percobaan dengan gerakan berbeda robot berhasil menjepit kok, melepas kok dan membalikkan servis. Hal-hal yang dapat mempengaruhi penurunan nilai tegangan dan tekanan udara ini diantaranya sebagai berikut.

- a. Terjadinya efek pembebanan pada rangkaian.
- b. Adanya toleransi komponen yang digunakan.
- c. Terjadi kesalahan pada saat pengukuran.
- d. Kondisi dari alat ukur yang kurang baik.
- e. Kondisi selang yang kurang rapat.
- f. Tekanan udara keluar saat piston keluar dan masuk.
- g. Kondisi silinder Pneumatik sudah lama digunakan.

C. Analisis pengujian motor

Motor pada robot ini berjumlah 5 buah, 4 buah motor berfungsi untuk mengatur arah jalan robot dan 1 buah motor untuk menggerakkan raket saat servis kok dan membalikkan servis lawan. Robot mampu bergerak cepat dan tepat.

Dari 5 kali percobaan, dapat dianalisis bahwa tegangan keluaran dari modul dapat memicu rangkaian elektronik pengendali motor dengan baik, sehingga robot mampu bergerak sesuai yang diinginkan. Nilai keberhasilan pengujian motor dapat dihitung sebagai berikut.

$$\frac{5}{5} \times 100\% = 100\%$$

5 kali percobaan dengan gerakan berbeda robot berhasil bermanuver, servis dan membalikkan servis. Hal-hal yang dapat mempengaruhi kecepatan dan gerak motor ini di antaranya sebagai berikut.

- a. Terjadinya efek pembebanan pada rangkaian.
- b. Adanya toleransi komponen yang digunakan.
- c. Kondisi lamanya motor digunakan sebelumnya.
- d. Tegangan dan arus yang mulai melemah.

D. Analisis pengujian komunikasi *wireless* robot

Aplikasi *wireless* pada robot ini berfungsi untuk mengontrol seluruh gerakan robot dari jarak jauh. Analisis data 3 kali percobaan, data yang dikirimkan dan diterima berupa data serial melalui bluetooth seri hc-05. Data yang diterima oleh sismin ATmega128 diproses untuk menggerakkan aktuator. Robot mampu bergerak sesuai yang diinginkan dalam jarak 1 meter hingga 13 meter, namun saat jarak >13 meter koneksinya jelek. Nilai keberhasilan komunikasi nirkabel dapat dihitung sebagai berikut.

$$\frac{13}{16} \times 100\% = 81,25\%$$

3 kali percobaan dengan jarak yang sama terjadi putus komunikasi 3 kali. Kebutuhan jarak komunikasi sudah terpenuhi yakni 6,7 meter sesuai ukuran lapangan bulutangkis untuk satu sisi. Hal-hal yang dapat mempengaruhi komunikasi nirkabel ini di antaranya sebagai berikut.

- a. Jarak antar *bluetooth*, semakin jauh jaraknya maka koneksi akan semakin jelek.
- b. Data serial yang dikirim dan diterima, data yang diterima ataupun dikirim bisa berubah dikarenakan faktor mikrokontroler atau hardware yang sudah mulai berkurang nilai fungsinya.
- c. Adanya gangguan atau benda yang menghalangi terhubungnya komunikasi antar *bluetooth*.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Setelah melakukan pengamatan dan pembahasan pada Proyek Akhir yang berjudul sistem kendali nirkabel robot bulutangkis berbasis mikrokontroler, dapat diambil kesimpulan yaitu:

1. pembuatan sistem kendali nirkabel robot bulutangkis berbasis mikrokontroller meliputi:
 - a. identifikasi kebutuhan yang diperlukan,
 - b. analisis kebutuhan untuk mendapatkan komponen secara spesifik,
 - c. perancangan perangkat keras dan perangkat lunak,
 - d. pembuatan alat meliputi: mekanik casis, mekanik utama robot, sistem minimum ATmega16, sistem minimum ATmega128 dan catu daya,
 - e. menentukan algoritma dengan membuat diagram alir/*flowchart*,
 - f. pengujian dan pengambilan data.
2. unjuk kerja sistem kendali nirkabel robot bulutangkis berbasis mikrokontroler meliputi:
 - a. tegangan catu daya dari regulator bernilai 5V mampu memenuhi kebutuhan sistem minimum ATmega16 dan ATmega128 dengan persentase keberhasilan 100%,
 - b. sistem minimum ATmega16 dan ATmega128 mampu mengendalikan pneumatik untuk menjepit kok, melepas kok dan membalikkan servis dengan persentase 100%,

- c. sistem minimum ATmega16 dan ATmega128 mampu mengendalikan motor untuk melakukan manuver, servis kok dan membalikkan servis dengan persentase keberhasilan 100%,
- d. sistem minimum ATmega16 dan ATmega128 mampu berkomunikasi dengan baik hingga jarak 13 meter dengan persentase keberhasilan 81,25% sudah memenuhi jarak maksimal arena bulutangkis satu sisi yakni 6,7 meter.

B. Keterbatasan Alat

Pembuatan sistem kendali nirkabel robot bulutangkis *wireless* berbasis mikrokontroler memiliki keterbatasan yaitu:

1. respon 4 buah motor base kurang cepat, ketika motor diperlambat masih terdapat sisa putaran sebelumnya,
2. menggunakan 4 buah motor lama pada base robot yang ternyata kecepatan antara motor kiri dan kanan sudah berbeda.

C. Saran

Berdasarkan kesimpulan dan keterbatasan yang telah dijelaskan, dapat disampaikan saran perlunya menggunakan 4 buah motor base dengan karakteristik kecepatan yang sama.

DAFTAR PUSTAKA

- Andra.(2014).Motor planetary gear . Diambil pada tanggal 21 Februari 2016 dari <http://www.brontoseno.com/produk/pg42mzy40geared-motor-series/>
- Bagus Hari Sasongko (2012). Pemrograman Mikrokontroler Dengan Bahasa C. Yogyakarta: Andi
- Bejo, Agus.(2008).C dan AVR Rahasia kemudahan bahasa C dalam mikrokontroler.yogyakarta:graha ilmu
- Digiware.(2014). Datasheet EMS 30 A H-Bridge . Diambil pada tanggal 21 Februari 2016 dari <http://digiwarestore.com/id/drivermodules/ems-30a-h-bridge-991256.html>
- Deddy Susilo. (2010). 48 Jam Kupas Tuntas Mikrokontroler MCS51 & AVR. Yogyakarta: Andi
- Edi Nur Rochman. (2008). Mikrokontroler AVR ATmega128/64.Diambil pada tanggal 21 Februari 2016 dari <http://edtronics.wordpress.com>.
- Eviandriani.(2005).Pengertian Sistem Kendali. Diambil pada tanggal 21 Februari 2016 dari <http://www.eviandriani.com/2010/05/pengertian-sistem-kendali.html>
- Fery Pratama. (2013). Robot Pemadam Api Berkaki Enam Berbasis Mikrokontroller ATmega16 Dan ATmega128. Yogyakarta: Proyek Akhir.
- Heri Andrianto. (2008). Pemrograman Mikrokontroller AT Mega 16 menggunakan Code Vision AVR. Bandung: Informatika.
- Iwan Setiawan. (2006). Tutorial Mikrokontroler AVR. Diambil pada tanggal 21 Februari 2016 dari <http://iwan@elektro.ft.undip.ac.id>.
- Khairudin, Moh.(2015). Sistem Kendali Ayunan Pada Robot Lengan RaketBerbasis Kombinasi *Proximity Sensor* Dan Pneumatik Untuk Menghasilkan Optimasi Pukulan *Shuttlecock*, Laporan Penelitian Departemen Pendidikan dan Kebudayaan UNY.
- Paulus Andi Nalwan. (2004). AN-0012 Jenis-jenis Motor. Diambil pada tanggal 21 Februari 2016 dari <http://www.robotindonesia.com>.
- Pragola, Dikka. (2013). Perancangan Dan Pembuatan *Mobile Robot* Dengan Sistem Navigasi Berbasis *Trajectory* Dan *Odometry*. Yogyakarta: Proyek Akhir.

Rule Aburobocon.(2015). Peraturanlomba robominton abu robocon 2015. Diambil pada tanggal 21 Februari 2016 dari <http://robocon.tvri.co.id/>

Robot Indonesia dan Kontes Robot Cerdas Indonesia).Yogyakarta: Graha Ilmu.

Widodo Budiharto dan Gamayel Rizal. (2007). 12 Proyek Mikrokontroler untuk Pemula. Jakarta: Elek Media Komputindo.

Widodo Budiharto. (2010). Robotika teori + implementasi. Yogyakarta: Andi.

Winarno & Deni Arifianto. (2011). *Bikin Robot Itu Gampang*. Jakarta: Kawan Pustaka

ATMEL. (2011). *ATmega128 Datasheet*. Atmel corporation

ATMEL. (2013). *ATmega16 Datasheet*. Atmel corporation

LAMPIRAN

Lampiran 2. Program sismin ATmega128

```

#include <mega128.h>
#include <delay.h>
#include <stdio.h>

void run_merah(); void run_biru();
void tampil_menu(void (*menunya)(unsigned char no_menu,unsigned
char jenis),unsigned char jumlah_menu);
void menu_utama(unsigned char no_menu,unsigned char jenis);
unsigned int n;
unsigned char pwm_raket;
unsigned char eRR,sensor,f1,f11=0;
int spd1,spd2,spd3,spd4;
unsigned char w1,w2,w3,w4,w5;

char mod;
// Alphanumeric LCD functions
#include <alcd.h>
unsigned char m_1,m_2,m_3,m_4,kec_m1,kec_m2,kec_m3,kec_m4;
//unsigned char pwm_1,pwm_2,pwm_3,pwm_4;
unsigned char buffer[30];

// External Interrupt 4 service routine
interrupt [EXT_INT4] void ext_int4_isr(void)
{
// Place your code here
m_1++;
}

// External Interrupt 5 service routine
interrupt [EXT_INT5] void ext_int5_isr(void)
{
// Place your code here
m_2++;
}

// External Interrupt 6 service routine
interrupt [EXT_INT6] void ext_int6_isr(void)
{
// Place your code here
m_3++;
}

// External Interrupt 7 service routine
interrupt [EXT_INT7] void ext_int7_isr(void)
{
// Place your code here

```



```

m_4++;
}
#ifndef RXB8
#define RXB8 1
#endif

#ifndef TXB8
#define TXB8 0
#endif

#ifndef UPE
#define UPE 2
#endif

#ifndef DOR
#define DOR 3
#endif

#ifndef FE
#define FE 4
#endif

#ifndef UDRE
#define UDRE 5
#endif

#ifndef RXC
#define RXC 7
#endif

#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<DOR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)

// USART1 Receiver buffer
#define RX_BUFFER_SIZE1 8
char rx_buffer1[RX_BUFFER_SIZE1];

#if RX_BUFFER_SIZE1 <= 256
unsigned char rx_wr_index1,rx_rd_index1,rx_counter1;
#else
unsigned int rx_wr_index1,rx_rd_index1,rx_counter1;
#endif

// This flag is set on USART1 Receiver buffer overflow
//bit rx_buffer_overflow1;

```

```

// USART1 Receiver interrupt service routine
interrupt [USART1_RXC] void usart1_rx_isr(void)
{

    status=UCSR1A;
    data=UDR1;
    if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
    {

        if(data == 255)
        {
            rx_wr_index1 = 0;}
        else
        {
            rx_buffer1[rx_wr_index1]=data;
            rx_wr_index1++;
        }
    };
}

#pragma used+
char getchar1(void)
{
    char data;
    while (rx_counter1==0);
    data=rx_buffer1[rx_rd_index1++];
    #if RX_BUFFER_SIZE1 != 256
    if (rx_rd_index1 == RX_BUFFER_SIZE1) rx_rd_index1=0;
    #endif
    #asm("cli")
    --rx_counter1;
    #asm("sei")
    return data;
}
#pragma used-

// Write a character to the USART1 Transmitter

#pragma used+
void putchar1(char c)
{
    while ((UCSR1A & DATA_REGISTER_EMPTY)==0);
    UDR1=c;
}
#pragma used-

// SPI functions
#include <spi.h>

```

```

// Declare your global variables here

//=====deklarasi
PS=====//
#define psxatn PORTB.4

//=====deklarasi OUTPUT
MOTOR UTAMA=====//

#define m1      PORTF.6
#define m2      PORTF.7
#define m4      PORTF.2
/*
#define dir_m1   PORTF.5
#define dir_m2   PORTF.4
#define dir_m3   PORTD.3
#define dir_m4   PORTF.3
*/
//=====solenoid=====
=====//
#define orientasi      PORTD.4
#define gripper        PORTD.5
#define putar          PORTD.6
#define push_up        PORTD.0
#define push           PORTD.7

//===== Planetary
=====
#define direksiki      PORTA.6
#define direksika      PORTA.7
#define planetary      PORTD.6

char buf[2];

//=====deklarasi TOMBOL
PSX=====//

unsigned char a0,a1,a2,a3,a4,a5,a6,a7;
unsigned char b0,b1,b2,b3,b4,b5,b6,b7;
unsigned char byte4, byte5, byte6, byte7, byte8, byte9;
int count=1,tim1,tim2;
//int f,g,h,i,j,k,l;

#include "set.c"

//=====deklarasi motor kiri
kanan=====//
unsigned char pwm1,pwm2,pwm3,pwm4,pwm_pw;

```

```

unsigned char x_motor;

unsigned char x=0,y=0,y2=0,y3=0,timer=0;
//pwmmotor utama//
// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
    // Reinitialize Timer 0 value
    TCNT0=0xFF;
    // Place your code here
    x_motor++;
    x++;
    if(x==255)
    {y++; x=0;}

    if(pwm2>x_motor) PORTD.1=1; else PORTD.1 = 0;
    if(pwm3>x_motor) PORTD.0=1; else PORTD.0 = 0;
    if(pwm4>x_motor)
    {
        PORTA |= 0b00000001;
    }
    else
    {
        PORTA &= 0b11111110;
    }

    if(pwm_raket>x_motor) planetary = 1;
    else planetary = 0;

    y2++;

    if(y2 == 254)
    {
        y3++; y2 = 0;
    }

    if(y3 == 20)
    {
        kec_m1 = m_1;
        kec_m2 = m_2;
        kec_m3 = m_3;
        kec_m4 = m_4;
        m_1 = 0;
        m_2 = 0;
        m_3 = 0;
        m_4 = 0;
        y3 = 0;
    }
}

```

```

    }

}

interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
    // Reinitialize Timer1 value
    TCNT1H=0xD5D0 >> 8;
    TCNT1L=0xD5D0 & 0xff;
    // Place your code here

}

#include "pid.c"
#include "direksi.c"

// Declare your global variables here
//pneumatik//
void default_pos(void)
{
    gripper = 1;
    orientasi = 1;
    putar = 1;
    push_up = 1;
    push = 1;
}

void lcd(unsigned char x,unsigned char y,unsigned char data)
{
    unsigned char a,b,c;

    a = data/100;
    data = data%100;
    b = data/10;
    c = data%10;

    lcd_gotoxy(x,y);
    lcd_putchar(a+0x30);
    lcd_putchar(b+0x30);
    lcd_putchar(c+0x30);
}

void tampilkan( int z)
{
    int data;
    data=z/100;
    lcd_putchar(48+data);
    z%=100;
}

```

```

data=z/10;
lcd_putchar(48+data);
z%=10;
lcd_putchar(48+z);
}

```

```

void tampil2( int z)
{
int data;
data=z/10;
lcd_putchar(48+data);
z%=10;
lcd_putchar(48+z);
}

```

```

void tampil3(int z)
{
int data;
data=z/1000;
lcd_putchar(48+data);
z%=1000;
data=z/100;
lcd_putchar(48+data);
z%=100;
data=z/10;
lcd_putchar(48+data);
z%=10;
lcd_putchar(48+z);
}
//spi//

```

```

void readPSX()
{
    psxatn = 0;
    spi(0X01);
    spi(0X42);
    spi(0X00);
    byte4 = spi(0X00);
    byte5 = spi(0X00);
    byte6 = spi(0X00);
    byte7 = spi(0X00);
    byte8 = spi(0X00);
    byte9 = spi(0X00);
    psxatn = 1;
}

```

```

void ambil(void)
{

```

```

unsigned char data4,data5;
readPSX();
if(PINA.4 == 0)//no bluetooth
{
a0=(byte4 & 0b00000001);
a1=(byte4 & 0b00000010)>>1;
a2=(byte4 & 0b00000100)>>2;
a3=(byte4 & 0b00001000)>>3;
a4=(byte4 & 0b00010000)>>4;
a5=(byte4 & 0b00100000)>>5;
a6=(byte4 & 0b01000000)>>6;
a7=(byte4 & 0b10000000)>>7;
b0=(byte5 & 0b00000001);
b1=(byte5 & 0b00000010)>>1;
b2=(byte5 & 0b00000100)>>2;
b3=(byte5 & 0b00001000)>>3;
b4=(byte5 & 0b00010000)>>4;
b5=(byte5 & 0b00100000)>>5;
b6=(byte5 & 0b01000000)>>6;
b7=(byte5 & 0b10000000)>>7;
}
else
{
data4 = rx_buffer1[4]+1;
data5 = rx_buffer1[5]+1;
a0=(data4 & 0b00000001);
a1=(data4 & 0b00000010)>>1;
a2=(data4 & 0b00000100)>>2;
a3=(data4 & 0b00001000)>>3;
a4=(data4 & 0b00010000)>>4;
a5=(data4 & 0b00100000)>>5;
a6=(data4 & 0b01000000)>>6;
a7=(data4 & 0b10000000)>>7;
b0=(data5 & 0b00000001);
b1=(data5 & 0b00000010)>>1;
b2=(data5 & 0b00000100)>>2;
b3=(data5 & 0b00001000)>>3;
b4=(data5 & 0b00010000)>>4;
b5=(data5 & 0b00100000)>>5;
b6=(data5 & 0b01000000)>>6;
b7=(data5 & 0b10000000)>>7;
};}

void ss(void)
{
ambil();
lcd_gotoxy(4,0);
sprintf(buf,"%d",a0); lcd_puts(buf);

```

```

    sprintf(buf,"%d",a1); lcd_puts(buf);
    sprintf(buf,"%d",a2); lcd_puts(buf);
    sprintf(buf,"%d",a3); lcd_puts(buf);
    sprintf(buf,"%d",a4); lcd_puts(buf);
    sprintf(buf,"%d",a5); lcd_puts(buf);
    sprintf(buf,"%d",a6); lcd_puts(buf);
    sprintf(buf,"%d",a7); lcd_puts(buf);
    lcd_gotoxy(4,1);
    sprintf(buf,"%d",b0); lcd_puts(buf);
    sprintf(buf,"%d",b1); lcd_puts(buf);
    sprintf(buf,"%d",b2); lcd_puts(buf);
    sprintf(buf,"%d",b3); lcd_puts(buf);
    sprintf(buf,"%d",b4); lcd_puts(buf);
    sprintf(buf,"%d",b5); lcd_puts(buf);
    sprintf(buf,"%d",b6); lcd_puts(buf);
    sprintf(buf,"%d",b7); lcd_puts(buf);
}

void tampil()
{
    readPSX();
    lcd(0,0,byte8);          // x kanan. bawah=0
    lcd(0,1,byte9);          // y
    lcd(13,0,byte6);          // x
    lcd(13,1,byte7);          // y
    ss();
}

bit aa;
bit bb;
bit cc;
bit dd;

//motor//
void set_speed(int s1, int s2, int s3, int s4)
{
    if(s1>10)
    {
        acuan1=s1;
        direksiM1(cw);
        aa=0;
    }
    else if(s1<-10)
    {
        acuan1=0-s1;
        direksiM1(ccw);
        aa=1;
    }
}

```



```

}
else
{
    acuan1=0;
    direksiM1(break_gnd);
    //if(aa==0) direksiM1(ccw);
    //else direksiM1(cw);
}

if(s2>10)
{
    acuan2=s2;
    direksiM2(cw);
    bb=0;
}

else if(s2<-10)
{
    acuan2=0-s2;
    direksiM2(ccw);
    bb=1;
}
else
{
    acuan2=0;
    direksiM2(break_gnd);
    pk = 0;
    pt = 0;

    //if (bb==0) direksiM2(ccw);
    //else direksiM2(cw);
}

if(s3>10)
{
    acuan3=s3;
    direksiM3(cw);
    cc=0;
}
else if(s3<-10)
{
    acuan3=0-s3;
    direksiM3(ccw);
    cc=1;
}
else
{
    acuan3=0;

```

```

    direksiM3(break_gnd);
    pk1 = 0;
    pt1 = 0;

    //if(cc==0) direksiM3(ccw);
    //else direksiM3(cw);
}

if(s4>10)
{
    acuan4=s4;
    direksiM4(cw);
    dd=0;
}

else if(s4<-10)
{
    acuan4=0-s4;
    direksiM4(ccw);
    dd=1;
}
else
{
    acuan4=0;
    direksiM4(break_gnd);
    pk2 = 0;
    pt2 = 0;
    //if(dd==0) direksiM4(ccw);
    //else direksiM4(cw);

}

pid();
}

char a = 0;
char b = 0;
char c = 0;
char d = 0;
char e = 0;

unsigned char data_0, data_1, data_2, data_3;

void set_servo(unsigned char derajat) // function to control servo
1
{
    PORTA |= 0b10000000; // berikan output logic 1

```

```

    delay_us(60); // selama 600us

    for(n=0; n < derajat; n++)
    { // ditambah 10us X derajat yg diinginkan
        delay_us(1);
    };

    PORTA &= 0b01111111;

    //servo=0; // keluarkan output logic 0
    for(n=0;n<(180-derajat);n++)
        delay_us(1); // tunda selama 10 ms

};

void send_motor(void)
{
    putchar1(255);

    putchar1(data_0);
    putchar1(data_1);
    putchar1(data_2);
    putchar1(data_3);
    /*
    lcd(7,0,data_0);
    lcd(11,0,data_1);

    lcd(7,1,data_2);
    lcd(11,1,data_3);

    putchar1(pwm1);
    putchar1(pwm2);
    putchar1(pwm3);
    putchar1(pwm4);
    */

    if(PINA.4 == 0)
    {
        lcd_gotoxy(0,1);
        lcd_putsf("0");
    }
    else
    {
        lcd_gotoxy(0,1);
        lcd_putsf("1");
    }
}

```

```

    lcd(7,0,rx_buffer1[1]);
    lcd(11,0,rx_buffer1[2]);

    lcd(7,1,rx_buffer1[3]);
    //lcd(11,1,rx_buffer1[4]+1);
    lcd(11,1,rx_buffer1[5]+1);

}
//joy stick//
#include "sevice.c"
void tombol(void)
{
    ambil();
    //lcd_gotoxy(0,0);

    data_0 = 0;
    //=====lengan=====//
    if((a7==0)&&(b3==1)) {data_1 = 10;}
    else if((a5==0)&&(b3==1)) {data_1 = 20;}
    else if((a7==0)&&(b3==0)) {data_1 = 30;}
    else if((a5==0)&&(b3==0)) {data_1 = 40;}
    else data_1 = 0;

    //=====lift=====//
    if((a4==0)&&(b3==1)) {data_2 = 30;}
    else if((a6==0)&&(b3==1)) {data_2 = 40;}
    else if((a4==0)&&(b3==0)) {data_2 = 50;}
    else if((a6==0)&&(b3==0)) {data_2 = 60;}
    else data_2 = 0;

    data_3 = 0;

    send_motor();

    ////////////GRIPER pneumatik
    ATAS////////////////////////////////////
    // if((b7==0)&&(b2==1)) { while(b7==0) {ambil(); gripper = 0;
    pendorong = 0;} b = 1; c = 1; }
    //segitiga
    if((b0==0)&&(a==0)) { while(b0==0) {ambil(); gripper = 0;}
    a = 1; }
    else if((b0==0)&&(a==1)){ while(b0==0) {ambil(); gripper = 1;}
    a = 0; }
    //bulat
    if(b7==0){direksiki=0;direksika=1;pwm_raket=50;f1 = 0;}
    else if(b6==0 && b2 == 1){direksiki=1;direksika=0;pwm_raket=255;
    f1 = 0;}
    else if(b6==0 && b2 == 0 ){service_otomatis();a = 0;}
    else if(a4==0 && b2 == 0 ){service_otomatis2();}

```

```

else      {direksiki=1;direksika=1;pwm_raket=0; fl = 1;}

    if((b2==0)&&(a5==0))      {   while(a5==0)
{ambil();service_manual();}}

/*
    if((b5==0)&&(b==0) && (b2==1)) { while(b5==0) {ambil(); gripper
= 0;} b = 1; }
    else if((b5==0)&&(b==1) && (b2==1)) { while(b5==0) {ambil();
gripper = 1;} b = 0; }
*/
//x

/*if((b6==0)&&(c==0) && (b2==1)) { while(b6==0) {ambil(); putar
= 0;} c = 1; }
    else if((b6==0)&&(c==1) && (b2==1)) { while(b6==0) {ambil();
putar = 1;} c = 0; }
*/
//Kotak
    if((b1==0)&&(d==0)) { while(b1==0) {ambil(); gripper=1;} d = 1;
}
    else if((b1==0)&&(d==1)) { while(b1==0) {ambil();gripper=0;} d =
0; }

    if((b5==0)&&(e==0)) { while(b5==0) {ambil(); orientasi = 0; push
= 0;}}
    else {ambil();push = 1; orientasi = 1;}
    //if(b5==0){direksiKi=1;direksiKa=0;pwm_pw=speed_lengan2;}
    //else if(b4==0){direksiKi=0;direksiKa=1;pwm_pw=50;}
    //else {direksiKi=1;direksiKa=1;pwm_pw=0;}

}

//stick analog//
void motor(void)
{
int selisih,spdL,spdR;
unsigned char analog_kiri_x;
unsigned char analog_kiri_y;
unsigned char analog_kanan_x;

    if(PINA.4 == 0)
    {

```

```

ambil();
analog_kiri_x = byte8;
analog_kiri_y = byte9;
analog_kanan_x = byte6;
}
else
{
analog_kiri_x = rx_buffer1[1]+1;
analog_kiri_y = rx_buffer1[2]+1;
analog_kanan_x = rx_buffer1[3]+1;
}

if(analog_kiri_y == 128) {selisih = (analog_kanan_x/2)-64;}
else { selisih = (analog_kanan_x - 128)/2;}

spdL = 128-selisih;
spdR = 128+selisih;

if(b3==1) mod = 4;
else mod = 2;

spd1 = (((analog_kiri_x-128)-(spdL-analog_kiri_y))/mod);
spd2 = (((analog_kiri_x-128)+(spdR-analog_kiri_y))/mod);
spd3 = (((128-analog_kiri_x)-(spdL-analog_kiri_y))/mod);
spd4 = (((128-analog_kiri_x)+(spdR-analog_kiri_y))/mod);

if(spd1>254){spd1=254;} else if(spd1<-254) {spd1 = -254;}
if(spd2>254){spd2=254;} else if(spd2<-254) {spd2 = -254;}
if(spd3>254){spd3=254;} else if(spd3<-254) {spd3 = -254;}
if(spd4>254){spd4=254;} else if(spd4<-254) {spd4 = -254;}

set_speed(spd1,spd2,spd3,spd4);
}

void manual(void)
{
motor();
lcd(0,0,PORTF);
tombol();
}
//proxymiti//
void set_atas_naik(){
do {
data_2=30;
send_motor();}
while (s_atas==1); data_2=0; send_motor();
}

void set_atas_turun(){

```

```

do {
data_2=40;
send_motor();}
while (s_atas==1); data_2=0; send_motor();
}

void set_bawah(){
do {
data_2=40;
send_motor();}
while (s_bawah==1); data_2=0; send_motor();
}

#include "menu.c"
#include "run_merah.c"
#include "run_biru.c"

void main(void)
{
PORTA=0x10; //1111 1000
DDRA=0xEF;  //0000 0111

PORTB=0x00; //0000 0000
DDRB=0x17;  //0001 0111

PORTC=0x00;
DDRC=0x08;

PORTD=0x00; //0000 0000
DDRD=0xFF;  //1111 0111

PORTE=0x00;
DDRE=0x0C;

PORTF=0x00; //0000 0000
DDRF=0xFC;  //1111 1100

PORTG=0x00;
DDRG=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 86.400 kHz
// Mode: Normal top=0xFF
// OC0 output: Disconnected
ASSR=0x00;
TCCR0=0x05;
TCNT0=0xFF;
OCR0=0x00;

```

```

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// OC1C output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x01;
TCNT1H=0xD5;
TCNT1L=0xD0;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
OCR1CH=0x00;
OCR1CL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
TCCR2=0x00;
TCNT2=0xFF;
OCR2=0x00;

// Timer/Counter 3 initialization
// Clock source: System Clock
// Clock value: Timer3 Stopped
// Mode: Normal top=0xFFFF
// OC3A output: Discon.
// OC3B output: Discon.
// OC3C output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer3 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off

```



```

// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR3A=0x00;
TCCR3B=0x00;
TCNT3H=0x00;
TCNT3L=0x00;
ICR3H=0x00;
ICR3L=0x00;
OCR3AH=0x00;
OCR3AL=0x00;
OCR3BH=0x00;
OCR3BL=0x00;
OCR3CH=0x00;
OCR3CL=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
// INT3: Off
// INT4: On
// INT4 Mode: Any change
// INT5: On
// INT5 Mode: Any change
// INT6: On
// INT6 Mode: Any change
// INT7: On
// INT7 Mode: Any change
EICRA=0x00;
EICRB=0x55;
EIMSK=0xF0;
EIFR=0xF0;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x05;

ETIMSK=0x00;

// USART0 initialization
// USART0 disabled
UCSR0B=0x00;

// USART1 initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART1 Receiver: Off
// USART1 Transmitter: On
// USART1 Mode: Asynchronous
// USART1 Baud Rate: 9600

```

```

UCSR1A=0x00;
UCSR1B=0x98;
UCSR1C=0x06;
UBRR1H=0x00;
UBRR1L=0x11;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC disabled
ADCSRA=0x00;

// SPI initialization
// SPI Type: Master
// SPI Clock Rate: 86.400 kHz
// SPI Clock Phase: Cycle Half
// SPI Clock Polarity: High
// SPI Data Order: LSB First
SPCR=0x7F;
SPSR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTC Bit 0
// RD - PORTC Bit 1
// EN - PORTC Bit 2
// D4 - PORTC Bit 4
// D5 - PORTC Bit 5
// D6 - PORTC Bit 6
// D7 - PORTC Bit 7
// Characters/line: 16

lcd_init(16);

// Global enable interrupts

#asm("sei")

default_pos();
PORTF=0X00;

```

```
//start//

while(1)
{
    default_pos();
    ambil();
    tampil();
    if(b6 == 0) break;;
}

lcd_clear();

while (1)
{

    default_pos();
    tampil_menu(&menu_utama,6);

}
}
```

Lampiran 3. Program Sismin ATmega16

```
#include <mega16.h>

// Alphanumeric LCD functions
#include <alcd.h>
#define psxatn PORTB.4
int data1,data2,data3,data4,data5;

#ifndef RXB8
#define RXB8 1
#endif

#ifndef TXB8
#define TXB8 0
#endif

#ifndef UPE
#define UPE 2
#endif

#ifndef DOR
#define DOR 3
#endif

#ifndef FE
#define FE 4
#endif

#ifndef UDRE
#define UDRE 5
#endif

#ifndef RXC
#define RXC 7
#endif

#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<DOR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)

// USART Transmitter buffer
#define TX_BUFFER_SIZE 8
char tx_buffer[TX_BUFFER_SIZE];

#if TX_BUFFER_SIZE <= 256
unsigned char tx_wr_index,tx_rd_index,tx_counter;
#else
unsigned int tx_wr_index,tx_rd_index,tx_counter;
#endif

// USART Transmitter interrupt service routine
interrupt [USART_TXC] void usart_tx_isr(void)
```

```

{
if (tx_counter)
{
--tx_counter;
UDR=tx_buffer[tx_rd_index++];
#if TX_BUFFER_SIZE != 256
if (tx_rd_index == TX_BUFFER_SIZE) tx_rd_index=0;
#endif
}
}

#ifndef _DEBUG_TERMINAL_IO_
// Write a character to the USART Transmitter buffer
#define _ALTERNATE_PUTCHAR_
#pragma used+
void putchar(char c)
{
while (tx_counter == TX_BUFFER_SIZE);
asm("cli")
if (tx_counter || ((UCSRA & DATA_REGISTER_EMPTY)==0))
{
tx_buffer[tx_wr_index++]=c;
#if TX_BUFFER_SIZE != 256
if (tx_wr_index == TX_BUFFER_SIZE) tx_wr_index=0;
#endif
++tx_counter;
}
else
UDR=c;
asm("sei")
}
#pragma used-
#endif

// Standard Input/Output functions
#include <stdio.h>

// SPI functions
#include <spi.h>

// Declare your global variables here
char buf[2];
unsigned char a0,a1,a2,a3,a4,a5,a6,a7;
unsigned char b0,b1,b2,b3,b4,b5,b6,b7;
unsigned char byte4, byte5, byte6, byte7, byte8, byte9;

void lcd(unsigned char x,unsigned char y,unsigned char data)
{
unsigned char a,b,c;

a = data/100;
data = data%100;
b = data/10;
c = data%10;

lcd_gotoxy(x,y);

```

```

    lcd_putchar(a+0x30);
    lcd_putchar(b+0x30);
    lcd_putchar(c+0x30);
}
//=====spi//
void readPSX()
{
    psxatn = 0;
    spi(0X01);
    spi(0X42);
    spi(0X00);
    byte4 = spi(0X00);
    byte5 = spi(0X00);
    byte6 = spi(0X00);
    byte7 = spi(0X00);
    byte8 = spi(0X00);
    byte9 = spi(0X00);
    psxatn = 1;
}

void ambil(void)
{
    readPSX();
    a0=(byte4 & 0b00000001);
    a1=(byte4 & 0b00000010)>>1;
    a2=(byte4 & 0b00000100)>>2;
    a3=(byte4 & 0b00001000)>>3;
    a4=(byte4 & 0b00010000)>>4;
    a5=(byte4 & 0b00100000)>>5;
    a6=(byte4 & 0b01000000)>>6;
    a7=(byte4 & 0b10000000)>>7;
    b0=(byte5 & 0b00000001);
    b1=(byte5 & 0b00000010)>>1;
    b2=(byte5 & 0b00000100)>>2;
    b3=(byte5 & 0b00001000)>>3;
    b4=(byte5 & 0b00010000)>>4;
    b5=(byte5 & 0b00100000)>>5;
    b6=(byte5 & 0b01000000)>>6;
    b7=(byte5 & 0b10000000)>>7;
}

void ss(void)
{
    ambil();
    lcd_gotoxy(4,0);
    sprintf(buf,"%d",a0); lcd_puts(buf);
    sprintf(buf,"%d",a1); lcd_puts(buf);
    sprintf(buf,"%d",a2); lcd_puts(buf);
    sprintf(buf,"%d",a3); lcd_puts(buf);
    sprintf(buf,"%d",a4); lcd_puts(buf);
    sprintf(buf,"%d",a5); lcd_puts(buf);
    sprintf(buf,"%d",a6); lcd_puts(buf);
    sprintf(buf,"%d",a7); lcd_puts(buf);
    lcd_gotoxy(4,1);
    sprintf(buf,"%d",b0); lcd_puts(buf);
    sprintf(buf,"%d",b1); lcd_puts(buf);
    sprintf(buf,"%d",b2); lcd_puts(buf);

```

```

    sprintf(buf,"%d",b3); lcd_puts(buf);
    sprintf(buf,"%d",b4); lcd_puts(buf);
    sprintf(buf,"%d",b5); lcd_puts(buf);
    sprintf(buf,"%d",b6); lcd_puts(buf);
    sprintf(buf,"%d",b7); lcd_puts(buf);

}

void tampil()
{
    readPSX();
    lcd(0,0,byte8);          // x kanan. bawah=0
    lcd(0,1,byte9);          // y
    lcd(13,0,byte6);         // x
    lcd(13,1,byte7);         // y
    ss();
}

void kirim()
{
    ambil();
    data1 = byte8-1;
    data2 = byte9-1;
    data3 = byte6-1;
    data4 = byte4-1;
    data5 = byte5-1;

    if(data1 <= 1) data1 = 0;
    if(data2 <= 1) data2 = 0;
    if(data3 <= 1) data3 = 0;
    if(data4 <= 1) data4 = 0;
    if(data5 <= 1) data5 = 0;

    putchar(255);
    putchar(123);
    putchar(data1);
    putchar(data2);
    putchar(data3);

    putchar(data4);
    putchar(data5);

    lcd(7,0,byte8);
    lcd(11,0,byte9);

    lcd(7,1,byte6);
    lcd(11,1,byte7);

    lcd(0,0,byte4);
    lcd(0,1,byte5);
}

void main(void)
{
    // Declare your local variables here

```

```

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out
Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0
State0=0
PORTB=0x00;
DDRB=0xFF;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTD=0x00;
DDRD=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=0xFF
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;

```



```

TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: Off
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 38400

UCSRA=0x00;
UCSRB=0x48;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x13;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC disabled
ADCSRA=0x00;

// SPI initialization
// SPI Type: Master
// SPI Clock Rate: 86.400 kHz
// SPI Clock Phase: Cycle Half

```

```

// SPI Clock Polarity: High
// SPI Data Order: LSB First
SPCR=0x7F;
SPSR=0x00;

// TWI initialization
// TWI disabLED
TWCR=0x00;

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTC Bit 0
// RD - PORTC Bit 1
// EN - PORTC Bit 2
// D4 - PORTC Bit 4
// D5 - PORTC Bit 5
// D6 - PORTC Bit 6
// D7 - PORTC Bit 7
// Characters/line: 16
lcd_init(16);

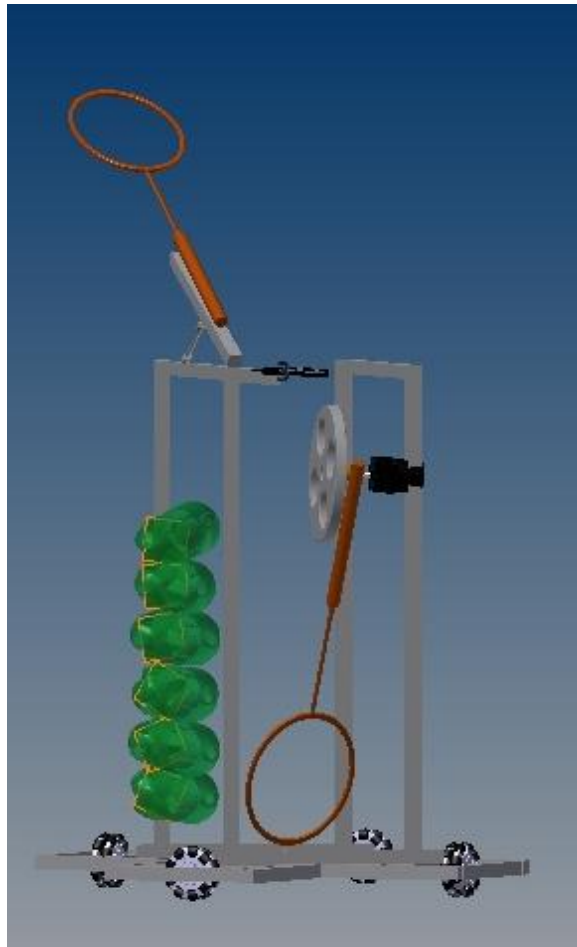
// Global enable interrupts
#asm("sei")

while(1)
{
    ambil();
    tampil();
    if(b6==0)break;;
}
    lcd_clear();
while (1)
{
    // Place your code here

    kirim();
}
}

```

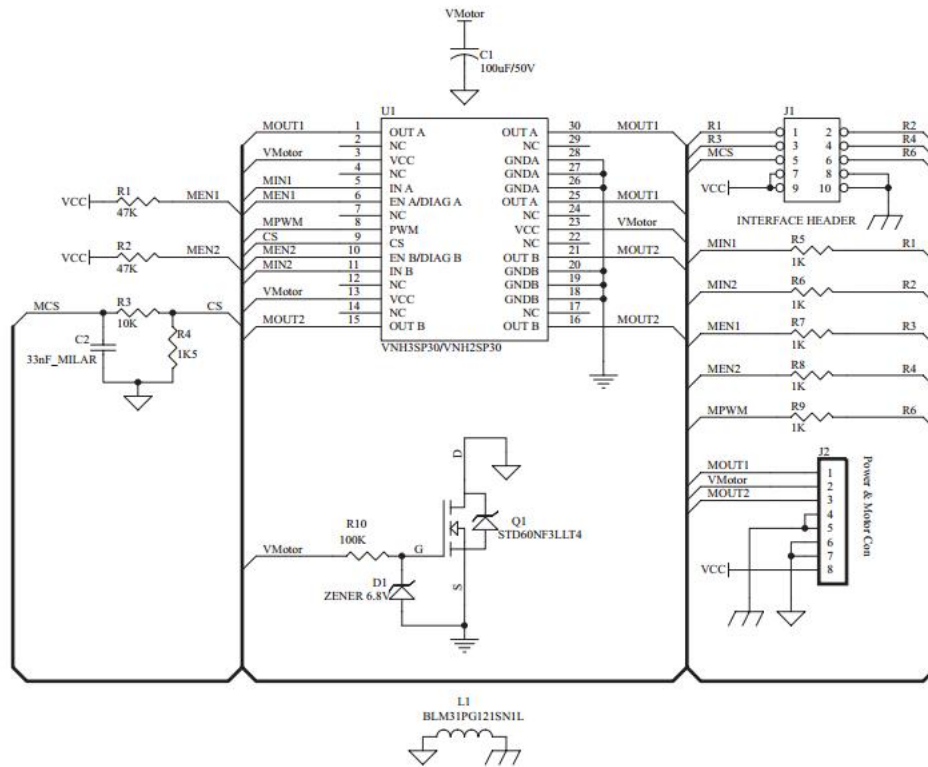
Lampiran 4. Desain mekanik robot



Lampiran 5. Daftar Komponen

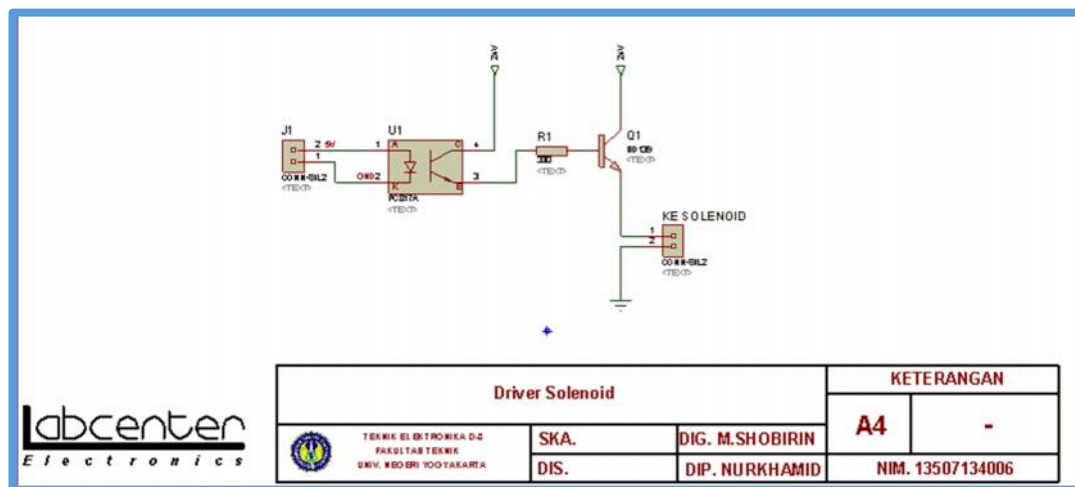
No	Material	Kuantitas
1	LCD 16x2	1 buah
2	ATmega16 dan ATmega128	2 buah
3	Driver Motor H-Bridge	5 buah
4	Bluetooth HC 05	2 buah
5	Sistem minimum	2 buah
6	Base Aluminium	2 buah
7	Kabel konektor	4set
8	Catu daya	2 buah
9	Driver Solenoid	1 set
10	Joystick	1 buah
11	Racket	2 buah
12	Pneumatik	2 buah
13	Botol tekanan tinggi	1 set
14	selang	1 set
15	Motor PG 45	5 buah
16	Roda omni	4 buah

Lampiran 6. Rangkaian elektronik penggerak motor H-bridge

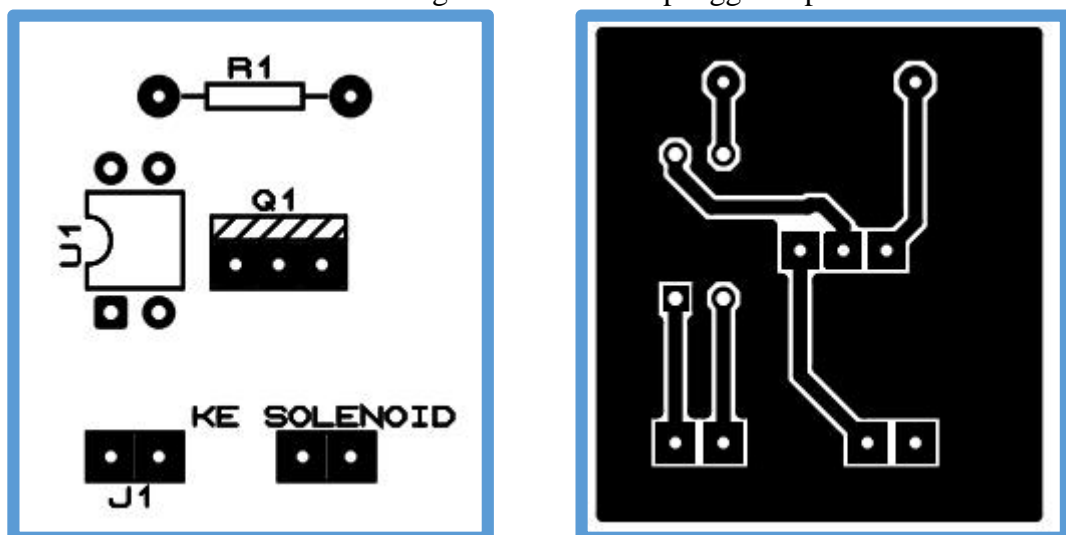


. Rangkaian elektronik penggerak motor H-bridge

Lampiran 7. Rangkaian elektronik penggerak pneumatik (driver pneumatik)



Gambar skematik rangkaian elektronik penggerak pneumatik



Gambar layout nampak atas dan nampak bawah rangkaian elektronik penggerak pneumatik